



Proyecto de Sistemas Informáticos

PROCESAMIENTO DE IMÁGENES ESTEREOSCÓPICAS EN ENTORNOS FORESTALES

Alumna

Cristina Sousa Real

Dirigido por:

Prof. Gonzalo Pajares Martinsanz.

Dpto. Ingeniería del Software e Inteligencia Artificial.

Curso 2008-09

AUTORIZACIÓN

Autorizo a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a la autora, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado

Fdo.: Cristina Sousa

Índice

| | |
|--|----|
| AUTORIZACIÓN | 3 |
| RESUMEN | 7 |
| ABSTRACT. | 7 |
| Capítulo 1. Introducción | 9 |
| 1.1 Generalidades | 9 |
| 1.2 Objetivos..... | 10 |
| 1.3 Motivación del proyecto | 11 |
| 1.4 Funcionalidad | 12 |
| 1.5 Organización de la memoria..... | 12 |
| Capítulo 2: Métodos de procesamiento | 13 |
| 2.1 Introducción..... | 13 |
| 2.2 Análisis de las imágenes: problemática..... | 13 |
| 2.3 Segmentación | 17 |
| 2.3.1 Segmentación basada en la umbralización por histograma | 18 |
| 2.3.2 Segmentación basada en regiones | 22 |
| 2.3.2.1 Algoritmo de crecimiento de regiones..... | 25 |
| 2.3.2.2 Algoritmo de División y fusión (“Split & Merge”)..... | 30 |
| 2.4 Correspondencia | 37 |
| 2.4.1 Problema de la correspondencia | 38 |
| 2.4.2 Métodos basados en el area | 44 |
| Capítulo 3: Diseño de la aplicación | 51 |
| 3.1 Módulos | 51 |
| 3.1.1 Procesamiento de imágenes..... | 52 |
| 3.1.2 Presentación al usuario | 52 |
| Capítulo 4: Interfaz de usuario | 57 |
| 4.1 Inicio | 57 |
| 4.2 Configurar..... | 58 |
| 4.3 Cargar imágenes | 59 |
| 4.4 Formulario | 62 |
| 4.5 Análisis | 64 |
| 4.5.1 Mostrar 3D..... | 66 |
| 4.5.2 Generar PDF | 66 |
| Capítulo 5. Conclusiones..... | 69 |
| 5.1 Resultados..... | 69 |
| 5.2. Observaciones..... | 72 |
| 5.3 Futuras líneas de desarrollo | 73 |
| Bibliografía..... | 75 |

RESUMEN

El proyecto realizado consiste en el desarrollo de una aplicación para el procesamiento de imágenes estereoscópicas capturadas con un sistema de dos cámaras con un tipo de lentes denominadas de ojo de pez. El objetivo final es la determinación del volumen de madera existente en un entorno forestal. La aplicación desarrollada se comunica con el usuario a través de un interfaz hombre-máquina amigable. El núcleo de la aplicación contiene una serie de métodos y algoritmos encaminados al procesamiento de las imágenes con el objetivo propuesto. Estos métodos se desencadenan a través de la mencionada interfaz mostrando en ella los resultados del procesamiento.

Palabras Clave: interfaz IHM, procesamiento imágenes, visión por computador, visión estereoscópica, lentes de ojos de pez, análisis forestal para madera.

ABSTRACT.

This project develops a specific application for stereoscopic image processing. The images are captured under the known fish eye lenses. The final goal is to compute the volume of wood existing in a forest environment. The application developed contains a friendly human interface. The kernel of the application contains several methods and techniques oriented toward the image processing under the proposed objective. These methods are triggered through the interface and the results obtained are displayed also in the interface.

Keywords: interface IHM, image processing, computer vision, stereoscopic vision, fish eye –lenses, wood forest analysis.

Capítulo 1. Introducción

1.1 Generalidades

El proyecto desarrolla una aplicación para la identificación de las texturas existentes en zonas boscosas con el fin último de estimar el volumen de madera existente en una determinada zona a partir de un sistema de visión estereoscópica basado en dos cámaras, que requiere a su vez el emparejamiento de los árboles en las dos imágenes del par estereoscópico. Las imágenes se captan mediante un sistema de lentes de ojos de pez, que tienen la capacidad de capturar la escena con ángulos de 360°.

Se parte por tanto, de dos imágenes obtenidas de la misma zona de árboles. Los métodos desarrollados en este proyecto procesan dichas imágenes tratando de identificar los árboles que aparecen en ellas, que a su vez son los existentes en la zona del bosque proyectada sobre las imágenes.

El proyecto consta de una interfaz de usuario, que conecta con los métodos y algoritmos de procesamiento de las imágenes a tratar. Como ya se ha mencionado previamente, al tratarse de un sistema estereoscópico son dos las imágenes que deben ser procesadas, debido a que como bien es sabido los sistemas estereoscópicos constan de dos cámaras, separadas éstas una cierta distancia entre ellas y por tanto, cada una de las cámaras proporciona una imagen diferente. El procesamiento de estas dos imágenes constituye la base para el planteamiento del proyecto.

A través de la interfaz, el usuario podrá elegir las imágenes en cuestión, lanzar los procesos necesarios y finalmente ver los resultados del análisis de manera sencilla, permitiéndole guardar los resultados del análisis, si así lo desea, en un fichero de texto en formato PDF.

Las técnicas implementadas para el procesamiento de las imágenes constituyen un subconjunto de todas las existentes en la literatura, que son ciertamente abundantes y en la mayoría de los casos bajo investigación. Por este motivo, en el presente trabajo no se incide tanto en los aspectos de investigación, por quedar fuera del alcance del proyecto, como de implementación, que constituye realmente el objetivo del desarrollo propuesto. Se ha hecho especial hincapié en los aspectos relativos al análisis, diseño e implementación bajo la interfaz de usuario de forma que el acceso a la aplicación se pueda realizar de una forma relativamente sencilla.

El lenguaje elegido para la implementación de la interfaz ha sido JAVA, entre otras razones por el hecho de disponer de un conjunto de librerías ya desarrolladas de gran utilidad. Entre los paquetes de librerías utilizados se encuentran el correspondiente a java3D, el de escritura de documentos en PDF o JAI (Java Advanced Imaging). Por otro lado, para desarrollar los algoritmos, que constituyen el núcleo de procesamiento se ha empleado Matlab, debido a que se trata de un paquete de toolboxes entre los que se incluye el Image Processing, específico para el tratamiento de imágenes. Además, las últimas versiones de MATLAB se han desarrollado en JAVA, por lo que resulta relativamente fácil la comunicación directa MATLAB-JAVA.

1.2 Objetivos

Como se ha comentado en el apartado anterior, el principal objetivo general del proyecto es poder analizar dos imágenes con el fin de distinguir automáticamente los árboles más cercanos y de ahí mediante los procesos apropiados determinar el volumen de madera de la zona analizada. No obstante, dado que la mayoría de los mencionados procesos se encuentran actualmente bajo investigación, los objetivos concretos de este trabajo se sintetizan en los siguientes:

1. Desarrollar una aplicación para el tratamiento de imágenes estereoscópicas con sistema de captura basados en lentes de ojos de pez.

2. Desarrollar una interfaz amigable de usuario para cargar las imágenes a tratar, seleccionar los métodos de procesamiento a aplicar y visualizar sobre dicha interfaz los resultados obtenidos.
3. Realizar un análisis y diseño de la aplicación bajo el punto de vista de las metodologías software.
4. Implementar una serie de métodos de procesamiento, integrando los mismos bajo la interfaz mencionada.
5. Realizar las pruebas pertinentes tanto individuales como de integración para verificar el correcto funcionamiento de la aplicación.
6. Extraer las conclusiones más relevantes en todas las fases de desarrollo de la aplicación, proporcionando las pautas necesarias para su continuidad en el futuro.

1.3 Motivación del proyecto

Este trabajo tiene su origen en el proyecto de investigación titulado “Análisis estereoscópico y reconocimiento de texturas en imágenes forestales obtenidas mediante dispositivo de medición MU-200501738”. Dicho proyecto se desarrolla dentro del grupo de investigación al que pertenece el director de este trabajo en colaboración con el Instituto Nacional de Investigación Agraria y Agroalimentaria (INIA) y la Escuela Técnica Superior de Ingenieros de Montes (ETSIM) de la Universidad Politécnica de Madrid siendo investigador principal del mismo el director de este trabajo de investigación.

La captura de las imágenes se lleva a cabo en el bosque objeto de estudio, situándose en diferentes áreas hasta completar la superficie total bajo análisis. Actualmente, las imágenes se procesan manualmente en los laboratorios de la ETSIM o del INIA para estimar el volumen de madera, estando en el ánimo de los organismos participantes el desarrollo de una aplicación lo más automatizada posible para disminuir el grado de intervención humana hasta donde se pueda, aumentando así la capacidad y potencialidad de análisis de imágenes, por el hecho de tener la posibilidad de procesar un mayor número de ellas en el mismo tiempo que el usuario.

1.4 Funcionalidad

Además del procesamiento de las imágenes y la obtención de las características de cada árbol (anchura, altura, distancia, ángulo), la aplicación le permite al usuario incluir datos adicionales que posteriormente resultarán de utilidad, como son:

- Lugar donde fueron tomadas las fotografías.
- Latitud y longitud del lugar.
- Fecha en que fue tomada y procesada la imagen.
- Comentarios que quiera agregar

Por otra parte, también se le permite al usuario guardar los resultados del procesamiento y los datos que ha añadido en el formulario, para futuras comparaciones o análisis posteriores. La aplicación está dotada con la capacidad de realizar una simulación de las imágenes procesadas para su visualización en tres dimensiones.

Además, a través del interfaz, se puede elegir entre varios algoritmos para poder comparar resultados y en su caso decidir qué procesamiento es el adecuado según la metodología utilizada.

1.5 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 2: Métodos de procesamiento: donde se describen los algoritmos y técnicas utilizadas para el procesamiento de las imágenes
- Capítulo 3: Diseño: en el que se incluyen los diagramas que representan el funcionamiento del sistema
- Capítulo 4: Interfaz de usuario: donde se explica cómo está organizada la interfaz de usuario.
- Capítulo 5: Conclusiones: En este apartado se incluyen los resultados de la aplicación, las conclusiones obtenidas y las futuras líneas para ampliar el proyecto.

Capítulo 2: Métodos de procesamiento

2.1 Introducción

En este capítulo se describen los métodos desarrollados relativos al procesamiento de las imágenes. Éstos se centran en dos aspectos fundamentales, que son los que se abordan como técnicas de procesamiento, a saber:

- Segmentación: consistente en el procesado previo de las dos imágenes por separado, consta a su vez de las dos siguientes partes
 - Segmentación por umbralización
 - Segmentación por detección de regiones
- Visión Estereoscópica: relativa al procesamiento conjunto de las dos imágenes del par estereoscópico, consta de los dos procesos siguientes
 - Algoritmos para resolver el problema de la correspondencia
 - Reconstrucción tridimensional de la escena

Antes de entrar en la descripción de los procesos anteriores, conviene realizar un análisis previo sobre la complejidad de las imágenes a tratar para determinar la problemática con la que nos enfrentamos.

2.2 Análisis de las imágenes: problemática

Este capítulo aborda la problemática estereoscópica para un tipo de sistemas característicos como son los formados por dos cámaras cuyas imágenes se obtienen mediante un tipo de lentes conocidas como de “ojo de pez”. En ellas el entorno tridimensional se proyecta de forma que se genera una imagen con proyección de los 360° alrededor de las cámaras. En las figuras 2.1 y 2.2 se muestran ejemplos representativos de imágenes capturadas con dichos sistemas.

Estos sistemas se caracterizan, además, porque las cámaras se sitúan entre ellas separadas una cierta distancia, en el caso que nos ocupa de un metro. Esto hace que un mismo objeto se proyecte en cada imagen en posiciones distintas. Obsérvese a este respecto en la figura 2.1 cualquiera de los troncos de los árboles, un mismo tronco aparece en una imagen en una determinada posición y en la otra imagen del par estereoscópico aparece girado un cierto ángulo con respecto al primero. Este hecho es el que permite determinar la estructura tridimensional de la escena, de suerte que es posible establecer la distancia a la que se encuentra el tronco en cuestión respecto del sistema de referencia de las cámaras. El objetivo del trabajo se centra precisamente, en determinar ese desplazamiento relativo para un mismo tronco entre las dos imágenes del par estereoscópico. Dicho desplazamiento se mide en grados. Dada la complejidad de este tipo de imágenes, como se verá posteriormente, el problema se desglosa en dos partes. En primer lugar, como el interés se centra exclusivamente en determinar las posiciones relativas de los troncos, es evidente que el resto de elementos presentes en la escena carecen de interés, este es el caso de las ramas de los árboles, el suelo o el cielo. Por ello, se enfoca el problema tratando de identificar aquellas texturas que son o no relevantes para el problema propuesto. Se trata de abordar esta problemática mediante técnicas de segmentación e identificación de texturas como las que se describen en apartados posteriores.

Una vez identificadas las texturas de interés y descartadas las que carecen de ello, el siguiente paso consiste en establecer las correspondencias entre los píxeles que constituyen los troncos de los árboles. Esta parte es la que coincide con el problema de la correspondencia en visión estereoscópica que se describe en el capítulo 2.4.

Por tanto, en este capítulo se describen las técnicas utilizadas, en algunos casos de forma independiente y en otros casos de forma combinada para mejorar los resultados obtenidos desde la perspectiva de segmentación de texturas y desde la correspondencia, siguiendo los métodos descritos en el capítulo dos.

Como ya se ha comentado previamente, el objetivo final del algoritmo consiste en aislar dentro de las imágenes exactamente los troncos de los árboles para poder extraer las propiedades anteriores con las que establecer la correspondencia en un sistema estéreo. Una vez establecida la correspondencia debe ser posible el cálculo del volumen de madera existente en las imágenes, cuestión ésta que no se aborda en el presente trabajo.

A la vista de las imágenes analizadas se identifica la problemática que a continuación se describe según los tipos de imágenes mostrados:

Imágenes captadas con escasa iluminación:

En este tipo de imágenes, que son las que corresponden a las de la figura 2.1, se observa la falta de iluminación, lo que se traduce en unas imágenes que han perdido la información de color, hecho éste que dificulta enormemente cualquier proceso posterior.

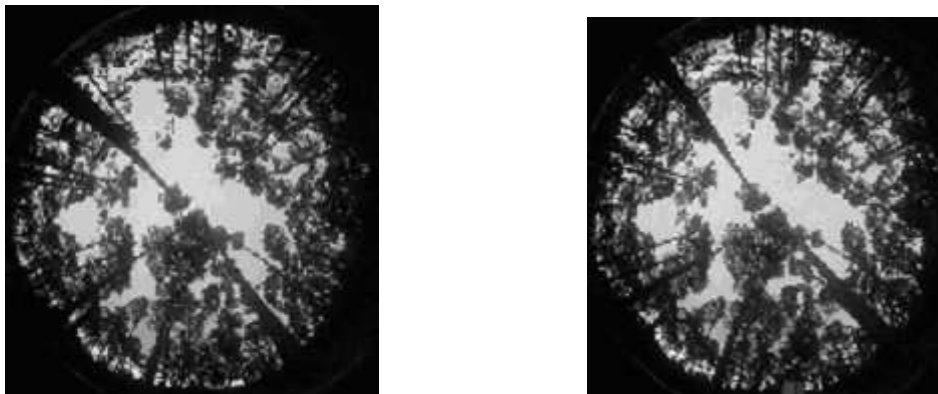


Figura 2.1 Par de imágenes estereoscópicas

En este caso, los troncos de los árboles presentan desde la base al menos tres tonalidades de color comenzando desde las negras y evolucionando hacia las copas de los árboles donde alcanza diferentes niveles o tonalidades (que resultan ser prácticamente grises). Esto implica que un mismo tronco de árbol no posee la misma textura de color debido a los diferentes niveles de iluminación.

Las tonalidades de los troncos son similares a las del resto de texturas existentes según la situación de la vegetación en la imagen.

Conclusión: no se pueden identificar los troncos sólo por la textura de color debido a su similitud con la maleza, cosa que no ocurre con las texturas del cielo que sí admiten una fácil discriminación

Descriptores de textura estadísticos:

En este tipo de imágenes es posible distinguir a través de los momentos estadísticos de segundo orden (varianza) diferentes texturas, diferenciándose las texturas de las ramas

de las de los troncos por presentar en este último caso una cierta variabilidad, mientras que en el primero se manifiestan de forma homogénea.

Imágenes con variedad de color

Existen otro tipo de imágenes, como las mostradas en la figura 2.2 con una apariencia totalmente distinta a la anterior. En este tipo de imágenes resulta ya más fácil separar la parte predominantemente verde de la que no lo es. Excluyendo las tonalidades azules, que corresponden al cielo, el resto representaría texturas de los troncos. Si bien, éstos presentan una problemática que es su distinta tonalidad; en efecto, se pueden distinguir varias tonalidades:

- a) Oscuras: quedan en la parte de la sombra
- b) Claras: por recibir la iluminación solar directamente, en este caso llega a tener el mismo valor que parte del cielo, con lo que la problemática respecto de su posible clasificación se incrementa considerablemente.

Estas dos apreciaciones se manifiestan en la imagen de la izquierda y no así en la imagen de la derecha, donde dado el espesor de vegetación no existen sombras y sólo cabe distinguir entre verde, azul, blanco y oscuro. Siendo la parte oscura la correspondiente a los troncos.

Se aprecian también diferentes texturas, distinguiéndose entre las texturas en las zonas donde existen hojas y las texturas de los troncos. Las primeras presentan alta variabilidad entremezclándose verde con azul, mientras que las texturas que representan los troncos no manifiestan este comportamiento.

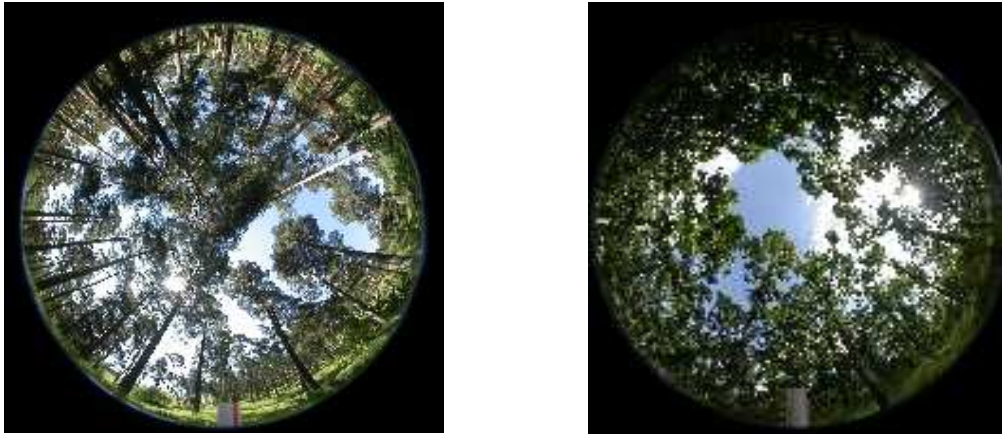


Figura 2.2 Imágenes captadas con sistema estéreo

2.3 Segmentación

Básicamente, en cualquier imagen se encontrarán presentes uno o varios objetos localizados en un entorno. El objetivo de la segmentación consiste en separar dichos objetos del medio en el que se encuentran y distinguirlos entre sí. En el caso de las imágenes objeto de la investigación este esquema no resulta tan simple, antes bien, presenta una enorme cantidad de texturas que dificultan su extracción, en cualquier caso, la segmentación, tanto en su versión simple como compleja se basa en los siguientes principios:

- Similitud: cada uno de los píxeles de un elemento tiene valores parecidos con respecto a alguna propiedad determinada.
- Discontinuidad: los objetos y más concretamente las regiones que los diferencian destacan del entorno y tienen por tanto unos bordes definidos.
- Conectividad: los píxeles pertenecientes al mismo objeto o región tienen que ser contiguos, es decir, tienden a agruparse constituyendo regiones homogéneas.

En nuestro caso, los elementos que queremos separar son los troncos de los árboles. Las imágenes que tenemos presentan ciertas dificultades debidas, en gran medida, a la iluminación. Los árboles no siempre poseen un color o textura que los diferencien fácilmente del resto de elementos. Debido también a la iluminación y al resto de elementos en la imagen, pueden aparecer regiones aparentemente no conectadas, esto es

aparecen troncos de árboles cortados, debido a la presencia de hierba u otros elementos sobre ellos.

Tras el proceso de segmentación se espera obtener una serie de regiones que definan los troncos cercanos al punto de vista. A continuación se describen una serie de métodos clásicos, que constituyen la base para realizar la segmentación.

2.3.1 Segmentación basada en la umbralización por histograma

El histograma de una imagen nos permite conocer la frecuencia relativa de aparición de cada uno de los posibles niveles de intensidad dentro de la imagen en cuestión. El histograma nos servirá para agrupar los píxeles en función de su nivel de intensidad luminosa. En la figura 2.3 podemos ver una imagen muy sencilla formada por un objeto oscuro sobre un fondo gris uniforme. Su histograma se muestra en la figura 2.4.

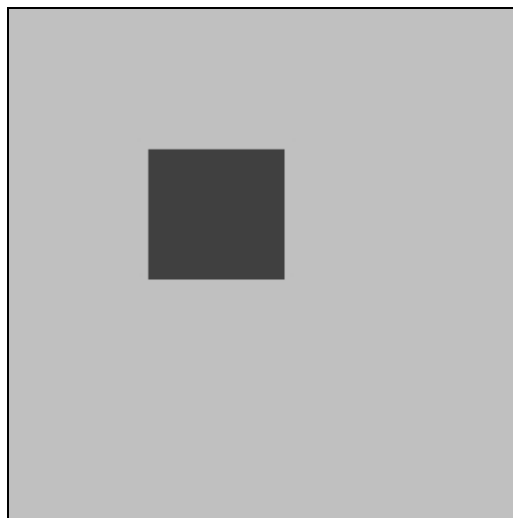


Figura 2.3 Imagen de color gris oscuro sobre un fondo gris claro.

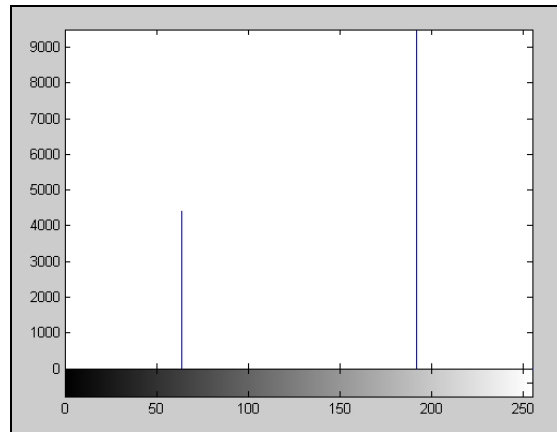


Figura 2.4 Histograma de la imagen de la Figura 2.3.

Puede verse en el histograma cómo aparecen dos líneas bien diferenciadas, que identifican las regiones existentes en la imagen, en este caso concreto y dada la imagen tan sencilla, se trata de un histograma bimodal. Aquí sería fácil calcular un umbral para distinguir las dos clases.

La umbralización basada exclusivamente en los niveles de luminosidad se denomina global, mientras que en el caso de la umbralización local se utiliza información puntual de cada píxel, es decir, propiedades de un píxel que dependen de su localización en la imagen.

Existen dos métodos básicos para abordar el problema de la umbralización global:

- Basada en la búsqueda de mínimos.
- Basada en las técnicas de reconocimiento de patrones.

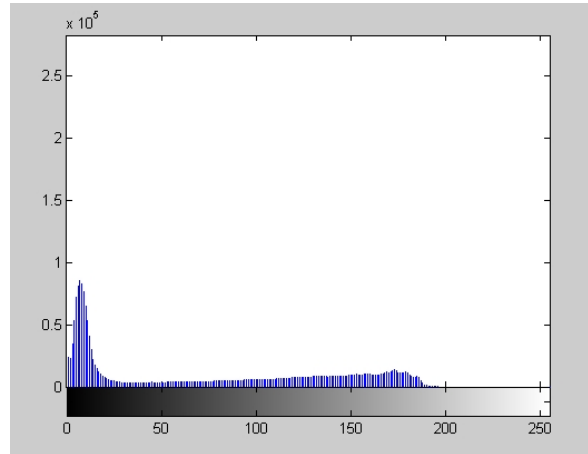
La umbralización basada en la búsqueda de mínimos trata de encontrar niveles de intensidad que separen el histograma de la imagen. Por otro lado, la umbralización basada en las técnicas de reconocimiento de patrones consiste, precisamente, en buscar patrones o formas en el histograma de la imagen. En nuestro caso, nos centraremos en el primer tipo de umbralización.

Este método funciona bien cuando los objetos buscados tienen un nivel de intensidad concreto que lo distingue del resto de objetos sobre los que se diferencian claramente.

En la figura 2.5 podemos ver una imagen forestal y su histograma de intensidad.



(a)



(b)

Figura 2.5 (a) Imagen forestal. (b) Histograma de (a).

A la vista del histograma de la figura 2.5 podemos encontrar fácilmente las dos clases que subyacen en la imagen. A la primera clase pertenecen niveles de intensidad entre 0 y 25 y a la segunda clase pertenecen niveles de intensidad entre 26 y 255. Aplicando esta división en dos clases obtenemos la imagen binaria de la figura 2.6.

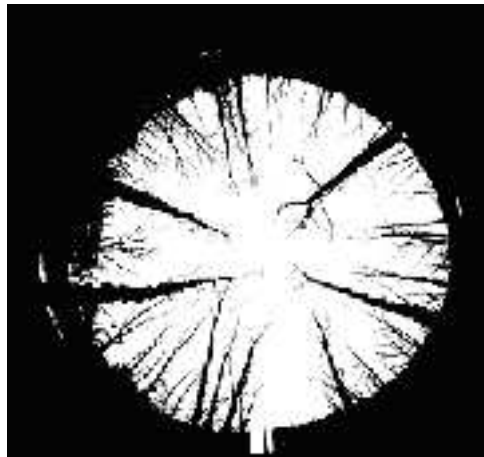


Figura 2.6 Imagen binaria obtenida por umbralización por histograma de la figura 2.5.

Esta división ha conseguido separar el cielo de los troncos. La siguiente fase consiste en recortar la parte de imagen perteneciente al suelo.

No siempre las imágenes son tan sencillas como la anterior. En la figura 2.7 se muestra otra de las imágenes forestales capturadas con su correspondiente histograma.

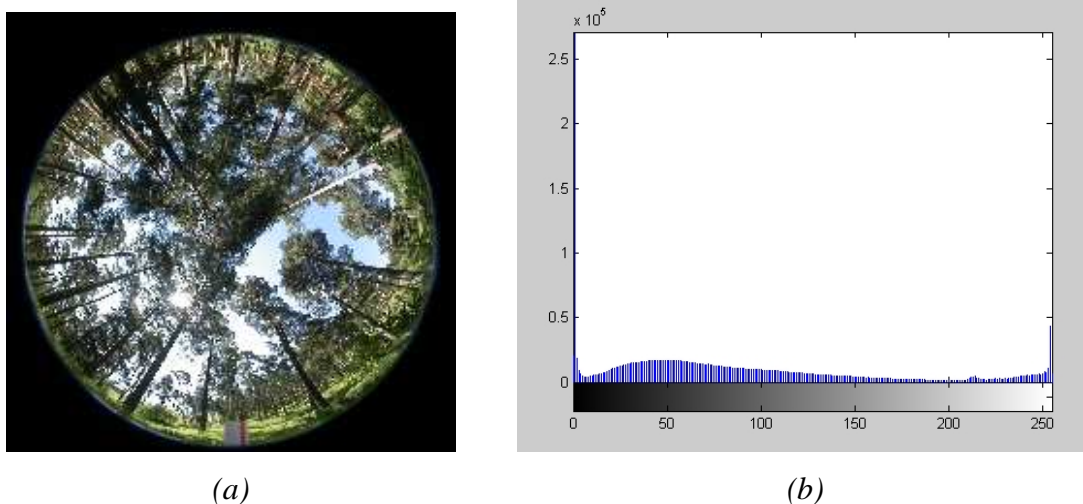


Figura 2.7 (a) Imagen de color compleja. (b) Histograma de (a).

En esta imagen las clases ya no aparecen tan diferenciadas como en el caso de las imágenes anteriores. Un posible umbral puede ser 75. Usando este umbral se obtiene la imagen binaria que aparece en la figura 2.8



Figura 2.8 Imagen binaria obtenida por umbralización de la figura 2.7(a).

La segmentación realizada ha conseguido capturar los troncos pero junto a los troncos han quedado partes de hojas. Esto se debe a que ambas clases de texturas poseen niveles de intensidad muy semejantes.

Si usamos el método de Otsu (1979), para calcular el valor umbral se obtiene un umbral de 94.98. En la figura 3.7 se muestra la imagen binaria obtenida tras la umbralización con este valor.

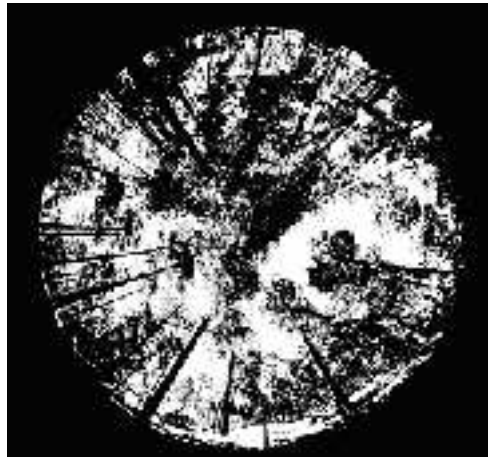


Figura 2.9 Imagen binaria obtenida por umbralización de la figura 2.7 con el método de Otsu.

Como podemos ver, el problema mencionado anteriormente persiste. Queda claro que en esta imagen no es suficiente con una umbralización por histograma. Será necesario aplicar métodos adicionales para eliminar los objetos que no pertenecen a troncos de árbol.

2.3.2 Segmentación basada en regiones

La segmentación basada en regiones busca dividir la imagen en particiones que comparten ciertas características comunes. En la figura 2.10 podemos ver una imagen en la que se han identificado tres regiones diferenciadas por su color. La región 1 pertenecería al fondo mientras que las regiones 2 y 3 pertenecerían a elementos de la imagen.

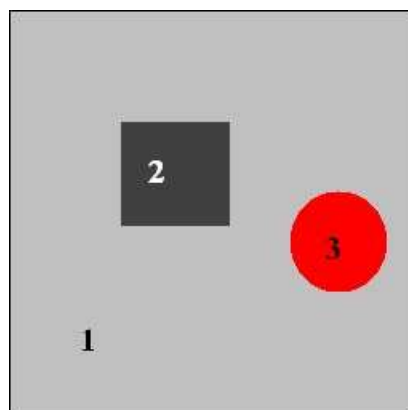


Figura 2.10 Imagen en la que aparecen dos objetos (2 y 3) sobre fondo gris claro.

El algoritmo general en la segmentación por detección de regiones consta de tres etapas:

1. Iniciar la partición
2. Definir un criterio de homogeneización de regiones
3. Optimizar la partición mediante modificaciones (división, unión, etc.). En algunos casos después de esta etapa se regresa de forma iterativa al paso 2.

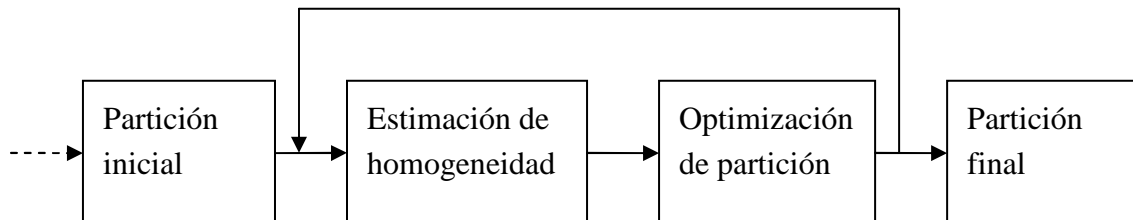


Figura 2.11. Etapas del algoritmo de segmentación basada en regiones.

1. Iniciar la partición. En esta primera etapa realizamos una primera segmentación muy simple. Una posible inicialización puede ser la existencia de una única región que engloba a toda la imagen, o justamente lo contrario, una región por cada píxel que forma la imagen. Esta inicialización también dependerá del conocimiento que tengamos sobre las regiones a extraer y de otros factores como puede ser la supervisión por parte de un experto que elija puntos clave para la posterior segmentación.
2. Criterio de homogeneidad. A la hora de segmentar una imagen debemos seleccionar unos criterios que nos permitan clasificar unas regiones como pertenecientes a una clase u otra. El color y las texturas son dos de estas características. Puede considerarse que un píxel forma parte de una región cuando comparte propiedades comunes con el resto de píxeles de dicha región. En ocasiones, la pertenencia o no a una región no debe tratarse a un nivel tan bajo como un píxel sino a un conjunto de ellos, como ocurre con las texturas de las regiones o los contornos que las definen.

Existen dos alternativas para aplicar el criterio de homogeneidad: determinista y aleatoria:

- Determinista:
 - Textura: Varianza respecto a un modelo.

$$C_T = \sum_n \sum_{i,j \in R_n} [x(i,j) - M_n(i,j)]^2 \quad (2.1)$$

Siendo R la región, (i, j) la posición y M el modelo con el que se quiere comparar.

- Contorno: minimizar la longitud para tener contornos simples. Un contorno simple es aquél que bordea la región objeto de estudio adaptándose a ella mediante la mínima longitud posible. Si podemos describir el contorno de una región podremos posteriormente comparar regiones a partir de estos descriptores de contorno.

C_c = Longitud de contornos

$$C = aC_t + (1-a)C_c \quad (2.2)$$

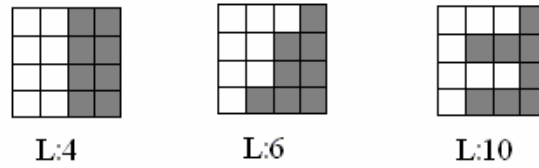


Figura 2.5 Ejemplo de cálculo de la longitud de Contornos.

- Aleatorio: estudio del contorno de configuraciones locales.

$$C = C_T \prod_{Transiciones} e^{(V_c)} \quad (2.3)$$

3. Modificaciones de la partición: en la última etapa del algoritmo, las particiones son modificadas y actualizadas para mejorar las condiciones de homogeneidad. Las principales modificaciones que se realizan en ésta fase consisten en la división y unión de regiones.

Dentro de la segmentación basada en regiones vamos a estudiar dos algoritmos clásicos:

1. Algoritmo de crecimiento de regiones.
2. Algoritmo de Fusión y División “Split & Merge”.

2.3.2.1 Algoritmo de crecimiento de regiones

La segmentación por crecimiento de regiones se basa en dos de los principios anteriormente mencionados: proximidad y similitud de puntos.

Para cada región que se desea segmentar, el crecimiento se inicia a partir de uno o más puntos iniciales, conocidos como semillas. A partir de este momento, la región va creciendo incorporando a aquellos vecinos inmediatos que satisfacen un criterio de aceptación prefijado. El crecimiento de esta región termina cuando no hay elementos que cumplan tal condición. Al finalizar el proceso tendremos un conjunto de regiones que engloban a un conjunto de puntos que la definen.

Selección de semillas

El principal problema en el caso del crecimiento de regiones es la ubicación de las semillas sobre la imagen de forma que a partir de esas semillas se generen las regiones. Por ello es importante que las semillas se especifiquen en posiciones de la imagen que corresponden a puntos de interés. La selección de las semillas puede hacerse de forma automática o con la ayuda de un experto que las seleccione de forma manual, que sería otra posible opción para que el método se orientara de una forma semiautomática.

Los puntos clave en este algoritmo son la selección de semillas y el criterio de aceptación de puntos dentro de una región, tal y como quedó patente en la exposición realizada previamente. Como ya se definió, una semilla es un punto a partir del que crece una región. Para la selección de semillas se han usado dos técnicas, una supervisada y otra no supervisada:

- Selección de semillas por un experto: en este caso, un experto se encarga de seleccionar los puntos clave a partir de los que se deben expandir las regiones. Estos puntos deben coincidir con un punto de cada tronco.
- Utilización de cada píxel como posible semilla: para cada punto en la imagen, que no ha sido etiquetado dentro de una región, se expande una nueva región a partir de él.

La primera opción tiene la desventaja de no ser un proceso totalmente automático. Sin embargo presenta un mejor rendimiento por no necesitar un recorrido exhaustivo de la

imagen. El segundo método necesita además alguna técnica adicional que permita conocer qué región calculada pertenece a un tronco y cuál no.

El segundo aspecto a tratar es el criterio de aceptación de un píxel dentro de una región. Para decidir si un punto pertenece a una región o no, sólo se han considerado los niveles de intensidad de dicho píxel, considerar también los vecinos de este píxel conlleva tiempos de cálculos demasiado elevados. El nivel de intensidad de dicho píxel puede compararse con el nivel de intensidad de la semilla solamente, con el nivel de intensidad del píxel vecino a partir del que se ha llegado o a partir del nivel de intensidad medio de la región.

Método supervisado de selección de semillas

En la figura 2.12 se muestran las regiones obtenidas utilizando la selección de semillas con el método supervisado para algunas de las imágenes capturadas. Podemos ver la semilla seleccionada y el resultado de realizar un crecimiento de región a partir de dicha semilla seleccionada de forma manual.

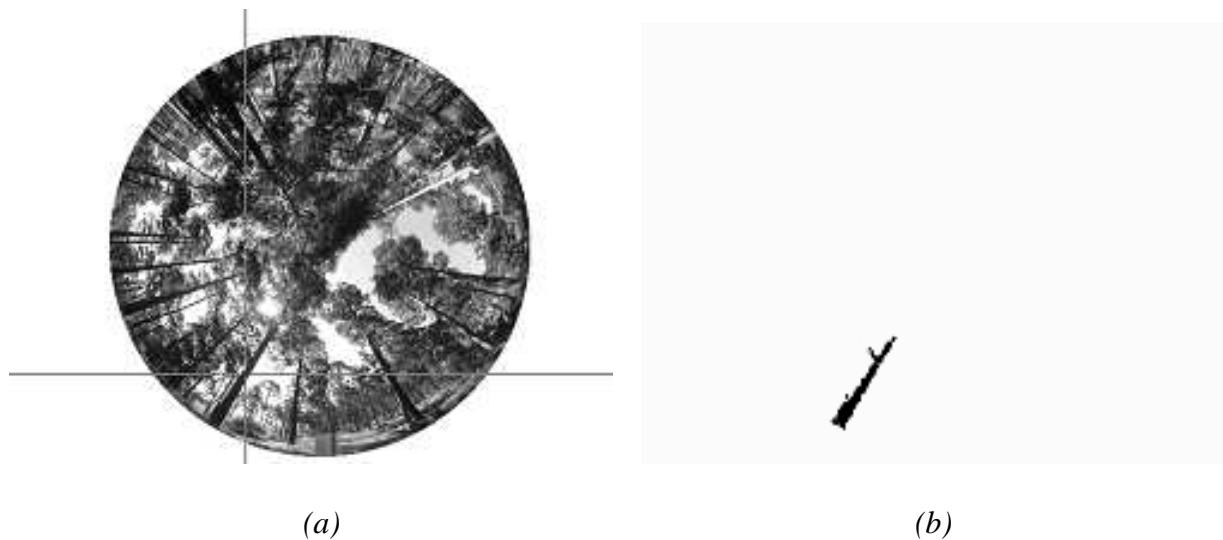


Figura 2.12 Segmentación supervisada. (a) Selección de semilla. (b) Región segmentada sobre región homogénea.

El problema que presenta esta técnica es que sólo funciona bien cuando toda la región del tronco es homogénea. Si el tronco presenta varias regiones sólo se obtendrá la región a la que pertenezca la semilla seleccionada. En la figura 2.13 podemos ver el

problema mencionado. Obsérvese en este sentido, el resultado, prácticamente inapreciable en la imagen de la parte (b).

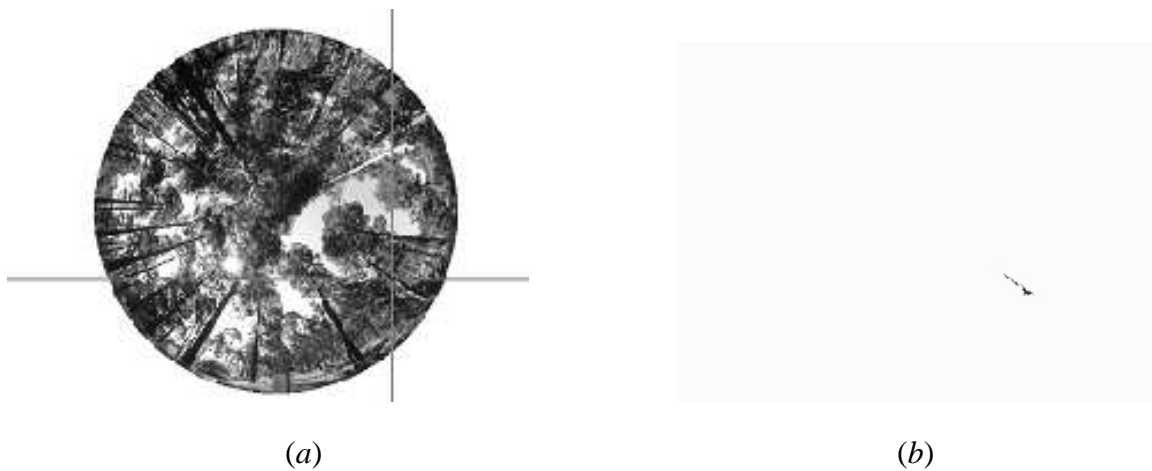


Figura 2.13 Segmentación supervisada. (a) Selección de semilla. (b) Región segmentada sobre región no homogénea.

Una posible solución a este problema es obligar al experto a seleccionar todas y cada una de las subregiones pertenecientes a un tronco. Este proceso puede ser excesivamente complejo y tedioso por lo que se ha optado por buscar un método automático de selección de regiones.

Método no supervisado de selección de semillas

Como ya se ha descrito, el proceso automático para buscar las regiones deseadas debe separar toda la imagen en regiones para después proceder a seleccionar aquellas que pertenecen a troncos de árboles. Para ello se recorre cada píxel en la imagen, si este píxel ha sido ya etiquetado en una región no se considera. Por el contrario si no ha sido etiquetado aún, entonces se crea una nueva región utilizando como semilla dicho píxel. En la figura 2.14 podemos ver el resultado obtenido, (b), para una imagen forestal (a). Un hecho destacable es que dicho resultado depende de la tolerancia permitida al aceptar un píxel como perteneciente a una región en función de la similitud del color utilizada.

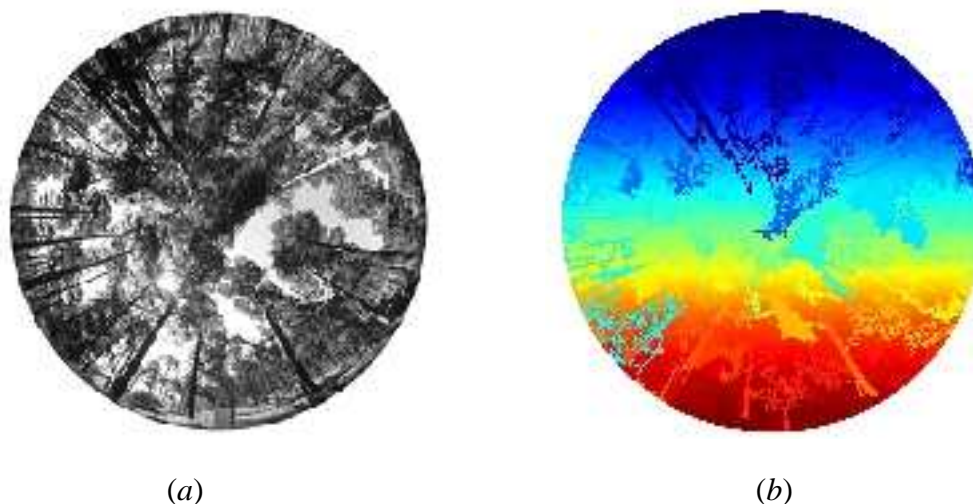


Figura 2.14 Segmentación no supervisada de regiones. (a) Imagen capturada. (b) Regiones extraídas en la segmentación.

Como puede observarse en la figura, algunas regiones han quedado bien separadas, sin embargo otras aparecen fusionadas con regiones formadas por hojas. El problema que presenta la segmentación de regiones forestales consiste en encontrar un umbral que nos permita discernir sobre si un píxel pertenece a una región u otra. Tras obtener estas regiones es necesario un criterio de aceptación de regiones como regiones propias de troncos. Entre los criterios que podemos usar para saber si una región pertenece a un tronco están:

- Nivel medio de intensidad que presenta la región.
- Forma de las regiones: los troncos deben ser regiones alargadas.
- Orientación: las regiones pertenecientes a troncos deben estar orientadas hacia el centro de la imagen.
- Área: puede usarse un área mínima como restricción para clasificar a una región como tronco o no.

Criterio de aceptación de los puntos

Se debe establecer un criterio apropiado para extraer las componentes de interés a partir del conjunto de puntos (píxeles) iniciales, definiendo las propiedades que dichos puntos deben cumplir para resultar incorporados a la región. Este criterio normalmente se basa en la proximidad y la homogeneidad de los píxeles adyacentes.

Por lo tanto, la elección del criterio a emplear para la evaluación de puntos candidatos constituye un aspecto crítico para el adecuado funcionamiento del algoritmo.

Un criterio simple de comparación se basa en el análisis del valor de intensidad del punto candidato con respecto al de las semillas especificadas, obteniendo una medida de distancia como sigue:

$$d(v) = |I(v) - \bar{I}(s)| \quad (2.4)$$

donde $I(v)$ es el valor de intensidad del punto candidato v e $I(s)$ se refiere a la intensidad de cada uno de los puntos s del conjunto S de semillas especificadas para la región. Además pueden considerarse los valores $d(v')$ correspondientes a los puntos v' en el entorno de v . Si para un determinado porcentaje p de los puntos evaluados se satisface que el valor d se encuentra dentro de un cierto límite de tolerancia t , el punto v se integra en la región, de lo contrario se descarta. Esta consideración permite lograr un criterio más robusto de crecimiento, el cual contribuye a prevenir situaciones de desbordamientos hacia otras componentes por conductos delgados y también problemas de sub-segmentación, generalmente debido al ruido. El hecho de considerar los puntos del tipo v' también conlleva un procesamiento mas costoso.

Un aspecto importante del algoritmo es la determinación de un valor de tolerancia para la diferencia de intensidades de los puntos de la región. El concepto de homogeneidad es bastante impreciso para el usuario, generalmente depende de las características de la imagen y varía según las distintas modalidades, por lo que no siempre es traducible efectivamente a un criterio computable, su especificación puede transformarse en un proceso de prueba y error.

Estrategia de crecimiento

La incorporación de puntos se basa en una estrategia de búsqueda en amplitud (Best First Search, BFS). Los puntos que son visitados y cumplen con la condición de aceptación son insertados en una lista hasta el momento de ser procesados y eventualmente integrados a la región. Inicialmente la lista está formada por las semillas. Luego se realiza el crecimiento en forma iterativa, extrayendo en cada ciclo el primer elemento u de la lista, el cual se etiqueta como perteneciente a la región para evitar su re-evaluación, y a partir de él se evalúan sus vecinos v en el entorno inmediato. Este

esquema asegura que cada punto se evalúa en una única oportunidad, aunque sí es posible que sea considerado más de una vez en el análisis del entorno de sus vecinos. El pseudocódigo de la figura 2.6 resume el esquema del algoritmo de segmentación por crecimiento de regiones.

```

Sea  $t$  la tolerancia establecida y  $p$  el porcentaje de vecinos similares exigidos
Insertar semillas en la lista  $L$ 
Mientras no se vacíe  $L$ 
    Sea  $u$  el primer elemento extraído de  $L$ 
    Rotular  $u$  como perteneciente a la región y marcarlo como visitado
    Calcular  $d(v)$ 
    Determinar los puntos  $v'$  en el entorno de  $v$  y calcular los  $d(v')$ 
    Si  $\left(\frac{n'}{n} \geq p\right)$ 
        Insertar  $v$  en  $L$ 
    Si no
        Descartar  $v$  y rotularlo como visitado

Siendo  $n'$  el número de puntos que se satisfacen que el valor  $d$  se encuentra dentro de
un cierto límite de tolerancia  $t$  y  $n$  el total de puntos evaluados.

```

Figura 2.15 Algoritmo de crecimiento de regiones

2.3.2.2 Algoritmo de División y fusión (“Split & Merge”)

Los aspectos en los que se basa el algoritmo de Split & Merge son la división y unión de regiones. Normalmente las dos fases del algoritmo se hacen precisamente en este orden, primero una división en regiones para posteriormente unir aquellas regiones vecinas similares.

Inicialmente la imagen es considerada como una única región cuadrada. Iterativamente, cada una de las regiones que componen la imagen son analizadas, comprobando si dicha

región es uniforme o no, según algún criterio previamente establecido, tal como que la diferencia de los valores de intensidad de los píxeles no deben diferenciarse entre sí más de un cierto valor respecto de uno central de referencia. Si es uniforme esta región se mantiene intacta. Si no es uniforme, la región es dividida en cuatro partes iguales que serán analizadas en las etapas posteriores. El proceso iterativo continúa hasta que las regiones son uniformes o lo suficientemente pequeñas.

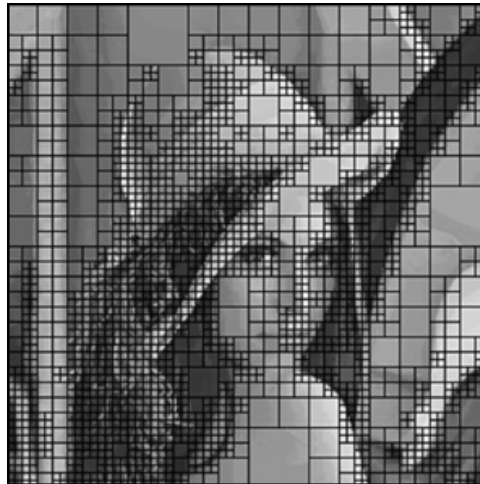


Figura 2.16 Imagen dividida en “quadtrees” mediante el algoritmo de División.

Cuando el proceso de división finaliza tiene lugar la fase de unión de las regiones obtenidas. En este caso, se ha de comprobar si dos bloques consecutivos, adyacentes, es decir, que estén en contacto espacialmente, cumplen el criterio anterior. Si es así, dichos bloques quedarán fusionados, formando un único bloque. Dicho proceso se repite hasta llegar a la idempotencia, es decir, el momento en el cual ya no pueden fusionarse más bloques.

El criterio adoptado en este proceso es igual que el aplicado en el anterior, con el apunte adicional de que el umbral de decisión puede ser variado entre la etapa de división y la de unión. Por lo general se suele tomar un umbral bastante estricto al realizar la división, suavizándolo a continuación en la fase de unión.

División

Se inicializa la lista de imágenes a procesar con la imagen completa.

Repetir hasta que la lista de imágenes a procesar esté vacía

- Extraer el primer elemento de la lista de imágenes a procesar
- Si la región cumple la condición de similitud entonces añadirla a la lista de regiones

Si no dividir la región en cuatro subregiones y añadirla a la lista de imágenes a procesar

Figura 2.17 Algoritmo División

Fusión

Añadir todas las regiones obtenidas en la etapa de división a la lista de imágenes a procesar.

Repetir hasta que no se produzca ninguna unión nueva

- Extraer cada región de la lista de imágenes a procesar y buscar una región vecina. Si al unir ambas regiones se sigue cumpliendo la condición de similitud entonces unir las y añadir la nueva región a la lista.

Figura 2.18 Algoritmo Fusión

El algoritmo de crecimiento de regiones es excesivamente sensible a variaciones en determinados píxeles. El algoritmo División y fusión considera texturas en vez de píxeles a nivel individual por lo que es menos sensible a cambios en los valores de intensidad o color al mismo nivel de píxel.

A la hora de diseñar el algoritmo de división debemos decidir el tamaño mínimo de ventana que se considera para tratar las texturas, junto con el criterio de homogeneidad de la región para llevar a cabo la división. Como criterio de homogeneidad se ha usado la desviación estándar y la media de color de la textura. Si la desviación no supera un cierto valor y la media se sitúa en un rango de valores se acepta la textura, en caso contrario se rechaza. En nuestro caso se han aceptado regiones con una desviación estándar no superior a 5 y una media de intensidad no superior a 150. De esta forma se eliminan las regiones con hojas y las regiones del cielo.

Se han realizado varias pruebas con distintos tamaños de ventana (N). En la figura 2.19 se muestran los resultados obtenidos usando División y Fusión (Split&Merge) con tamaños de ventana 3, 5, 7 y 11 sobre 2.19(a).

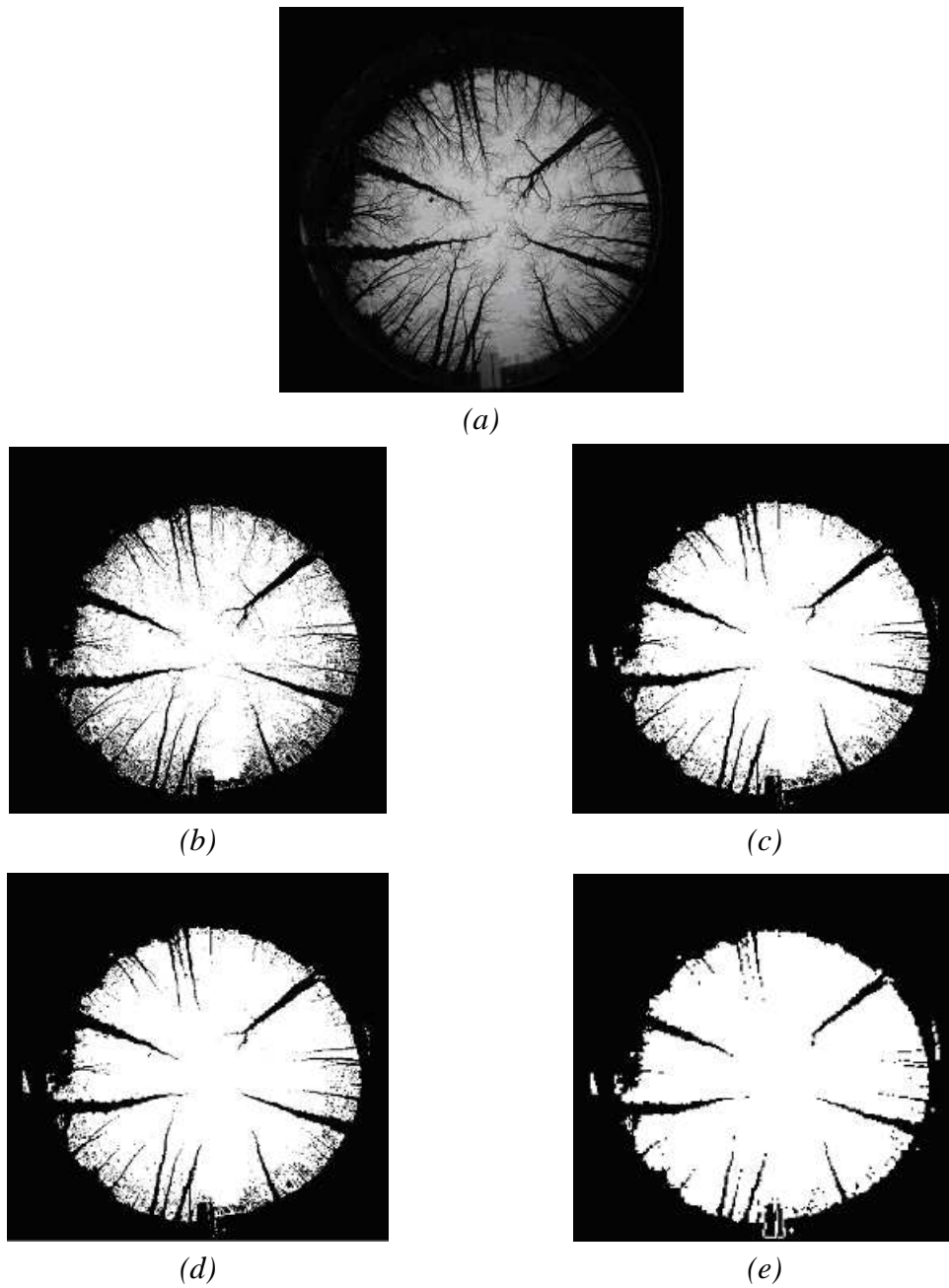


Figura 2.19 Algoritmo de Split&Merge usando distintos tamaños de ventana (N). (a) Imagen original. (b) Segmentación con $N=3$. (c) Segmentación con $N=5$. (d) Segmentación con $N=7$. (e) Segmentación con $N=11$.

Como se puede observar en la 2.19(b), un tamaño de ventana demasiado pequeño produce sobre-segmentación, mientras que con un tamaño de ventana grande podemos perder información tal y como se aprecia en la figura 2.19(e).

A partir de la imagen con tamaño de ventana 11 podemos eliminar las regiones con un área pequeña obteniendo la imagen mostrada en la figura 2.20.

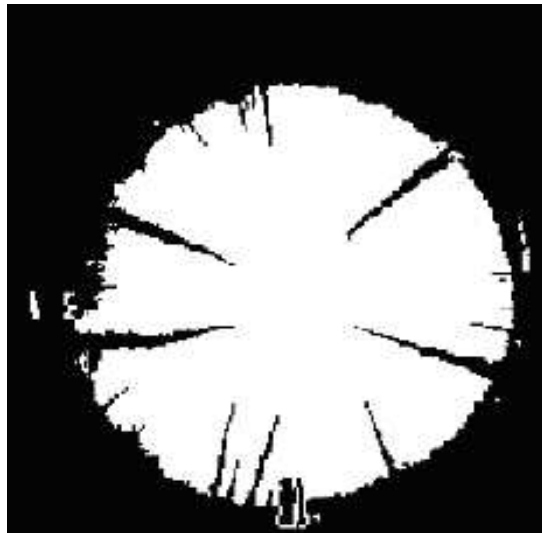


Figura .2.20 Segmentación eliminando regiones pequeñas

A partir del resultado obtenido en la figura 2.20 y eliminando lo que queda fuera de la circunferencia y las áreas resultantes con baja excentricidad llegamos al resultado de la figura 2.21.

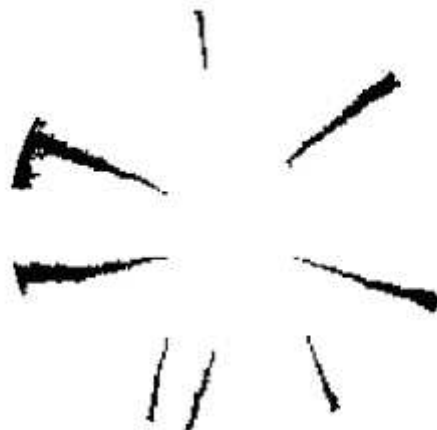
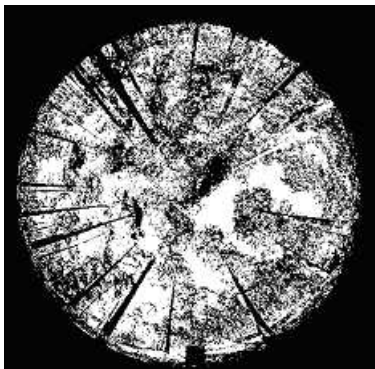


Figura 2.21 Segmentación eliminando suelo

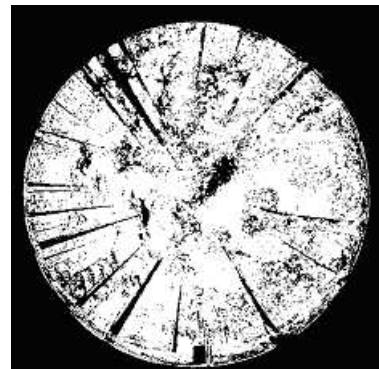
Tal y como se puede apreciar, el resultado obtenido sobre esta imagen resulta ser satisfactorio mediante la aplicación del método de segmentación expuesto. La dificultad se presenta cuando tenemos imágenes más complejas como la mostrada en la figura 2.22.



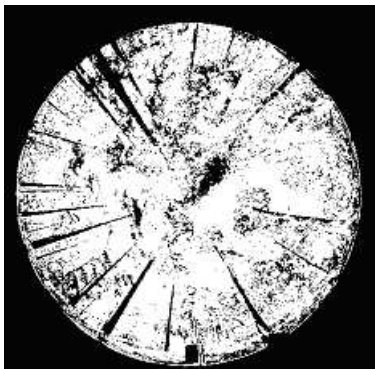
(a)



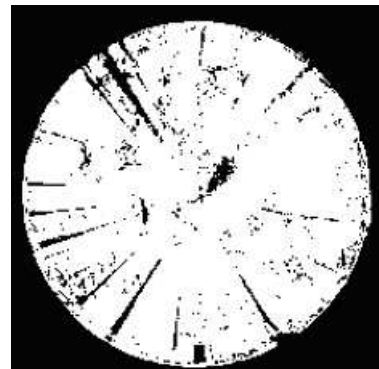
(b)



(c)



(d)



(e)

Figura 2.22 Algoritmo de Split&Merge sobre imagen compleja usando distintos tamaños de ventana (N). (a) Imagen original. (b) Segmentación con $N=3$. (c) Segmentación con $N=5$. (d) Segmentación con $N=7$. (e) Segmentación con $N=11$.

Podemos observar que en las imágenes segmentadas han quedado regiones que no son troncos y algunos troncos han desaparecido.

Si nos quedamos con la imagen 2.22(e), podemos mejorar la imagen eliminando regiones pequeñas y regiones con excentricidad alta. Esta operación se realiza mediante la extracción de propiedades de las regiones tales como el área y la excentricidad de las

mismas. Los experimentos realizados en este trabajo se han llevado a cabo con Matlab, que posee la función *regionprops*, con la que podemos obtener las propiedades mencionadas de área y excentricidad de cada una de las regiones



Figura 2.23 Segmentación eliminando regiones pequeñas

A pesar de haber conseguido una mejora en cuanto a los resultados de segmentación, todavía no se ha conseguido llegar a resultados satisfactorios, ya que se mantienen regiones que no son troncos de árboles y se han perdido otra serie de árboles que deberían haber permanecido.

Aplicando un cierto conocimiento de índole superior, se deduce que aquellas regiones pertenecientes a un tronco de árbol deben estar cerca del suelo, por lo que podemos eliminar aquellas regiones que se alejen del suelo, obteniendo en este caso el resultado que se muestra en la figura 2.24.



Figura 2.24 Segmentación eliminando regiones lejos del suelo

A la vista de los resultados obtenidos con este método de segmentación, podemos deducir que este algoritmo por sí mismo tampoco resulta ser satisfactorio para el proceso de segmentación que se pretende.

2.4 Correspondencia

La visión estereoscópica es el proceso por el cual se reconstruye un escenario 3D a partir de dos imágenes 2D. El cerebro humano percibe la realidad tridimensional a través del sistema de visión, que consta de dos ojos desplazados ligeramente uno respecto del otro según la dirección horizontal. Basándonos en este hecho los sistemas de percepción artificiales poseen dos cámaras desplazadas entre sí una cierta distancia, gracias a lo cual se puede recuperar la tercera dimensión perdida en las imágenes bidimensionales.

En la figura 2.25 se muestra un par de imágenes estereoscópicas obtenidas mediante dos cámaras desplazadas entre sí una cierta distancia horizontal.

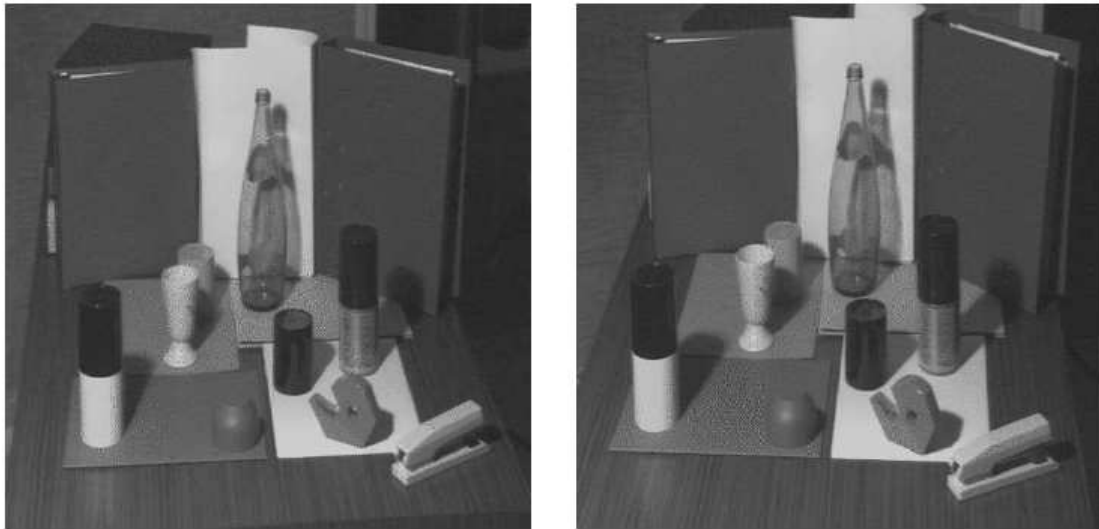


Figura 2.25 Imágenes capturadas por un sistema de visión estéreo

Para estimar la profundidad de dos imágenes estéreo debemos solucionar dos problemas principales:

- Problema de correspondencia: para todos los elementos en la imagen izquierda, encontrar sus elementos correspondientes en la imagen derecha, considerando ‘elementos’ tales como píxeles, características, regiones, objetos, etc.
- Problema de reconstrucción: usando las disparidades estimadas entre elementos, reconstruir la estructura 3D de la escena. Para esta etapa se necesita información adicional sobre las cámaras y hacer ciertas suposiciones sobre la escena. Esta etapa del proceso no se aborda en este trabajo.

2.4.1 Problema de la correspondencia

Dadas dos o más imágenes del mismo escenario 3D tomadas desde diferentes puntos de vista, el problema de la correspondencia consiste en encontrar conjuntos de puntos en una imagen que puedan ser identificados con los mismos puntos en la otra imagen. Un humano normalmente puede solucionar este problema fácilmente. En visión artificial el problema de la correspondencia se trata de solucionar de forma automática únicamente con las imágenes del par estereoscópico como entrada. Una vez que el problema de la correspondencia ha sido solucionado, obteniendo un conjunto de puntos de una imagen

con sus homólogos en la otra imagen, se pueden aplicar otros métodos para reconstruir la posición de dichos puntos en la escena 3D.

Métodos para establecer la correspondencia

Existen dos aspectos que deben ser considerados:

- Cómo seleccionar las correspondencias
- Cómo determinar la bondad de las correspondencias

En la literatura aparecen dos clases principales de algoritmos:

- Basados en el área: establecen la correspondencia atendiendo a los niveles de intensidad de la imagen píxel a píxel y teniendo en cuenta todos los píxeles de las imágenes.
- Basados en características: la correspondencia se establece mediante un conjunto de características de la imagen, tales como puntos de borde, puntos de interés, regiones, etc.

Con el problema de la correspondencia se trata de identificar cada píxel de la imagen izquierda en la imagen derecha.

En la figura 2.26 se muestra la correspondencia de un punto en la imagen izquierda con su homólogo en la imagen derecha. Como las cámaras que forman el sistema estéreo tienen perspectivas distintas, el mismo punto se proyecta en localizaciones diferentes.

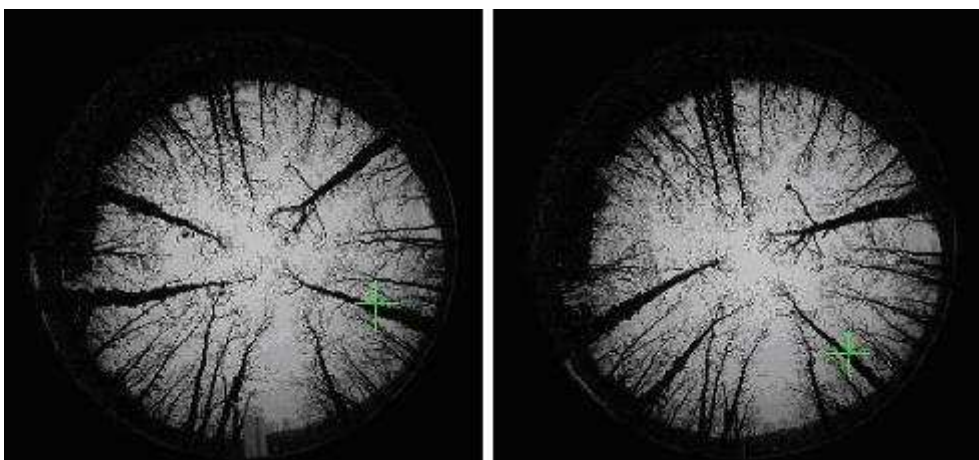


Figura 2.26 Problema de la correspondencia

Como ya se ha mencionado anteriormente, la correspondencia de imágenes es un proceso mediante el cual se trata de identificar la misma característica espacial en ambas imágenes del par estereoscópico. Desde un punto de vista humano, el problema resulta relativamente sencillo, ya que fijándose en un punto de una imagen resulta relativamente fácil identificar su homólogo en la otra imagen del par estereoscópico. Sin embargo, desde el punto de vista de la visión por computador el proceso se complica enormemente, ya que entre otras cosas, para un píxel o característica dada en una imagen pueden existir multitud de candidatos en la otra imagen. Para resolver esta problemática, hemos de abordar las cuestiones:

- ¿Qué elementos deben considerarse para la correspondencia?
- ¿Qué restricciones reducen la ambigüedad del proceso?

Los elementos más comúnmente empleados en la literatura general son:

- Píxel: podemos usar las características del píxel: nivel de gris, niveles de gris de la vecindad (ventana), conjunto de descriptores invariantes.
- Borde: se usan las características de segmentos rectilíneos: longitud, orientación, nivel gris medio.
- Píxel + borde.

Para abordar el problema de la correspondencia se utilizan lo que se conocen como restricciones estereoscópicas, destacando las siguientes: *similitud*, *unicidad*, *continuidad de la disparidad*, *epipolaridad*, (Pajares y Cruz, 2007). La similitud se refiere al hecho de que dado un punto en la escena, éste se proyectará con propiedades similares en la imagen, por ejemplo las hojas verdes de un árbol se proyectan en color verde en sendas imágenes. La unicidad indica que un punto en la escena se proyecta en un único punto en cada imagen (salvo oclusiones o fenómenos especulares). La continuidad de la disparidad indica que los píxeles de los objetos se situarán a distancias similares y en consecuencia a disparidad constante, por ejemplo los píxeles de un mismo tronco de árbol están todos ellos a la misma distancia del sistema estéreo y por tanto presentan la misma disparidad. Finalmente, la epipolaridad, se refiere a que las proyecciones de un mismo punto en la escena en sendas imágenes lo hacen siguiendo precisamente las denominadas líneas epipolares.

En el algoritmo de correspondencia que se presenta en este documento sólo se han tenido en cuenta las restricciones de similitud y epipolaridad. La similitud no puede ser garantizada completamente en nuestro caso debido a los efectos de iluminación.

Epipolaridad

En Romero (2008) se describe el proceso calibración de cámaras estereoscópicas con lentes de “ojo de pez” y las ecuaciones que definen la línea epipolar. La figura 3.19 representa la proyección de un punto objeto del espacio en cada una de las dos imágenes captadas por la cámara. El punto viene descrito por las coordenadas (x_1, y_1, z_1) y (x_2, y_2, z_2) en los sistemas de coordenadas de las cámaras 1 y 2 respectivamente. Se supone que la orientación de las cámaras es tal que comparten el eje Y y el plano XY. Además, α_1 y α_2 son los ángulos de incidencia del haz del punto objeto respecto al eje óptico de cada una de las cámaras; (x_{1i}, y_{1i}) y (x_{2i}, y_{2i}) son las coordenadas de los puntos imagen para cada cámara; los puntos (x_1, y_1, z_1) y (x_{1i}, y_{1i}) y el eje z_1 , pertenecen al mismo plano, al igual que ocurre con los puntos (x_2, y_2, z_2) y (x_{2i}, y_{2i}) y el eje z_2 . De acuerdo con lo expresado previamente, podemos establecer las siguientes relaciones (Schwalbe, 2005):

$$\begin{aligned} \frac{x_{1i}}{y_{1i}} &= \frac{x_1}{y_1} \\ \frac{x_{2i}}{y_{2i}} &= \frac{x_2}{y_2} \end{aligned} \quad (2.5)$$

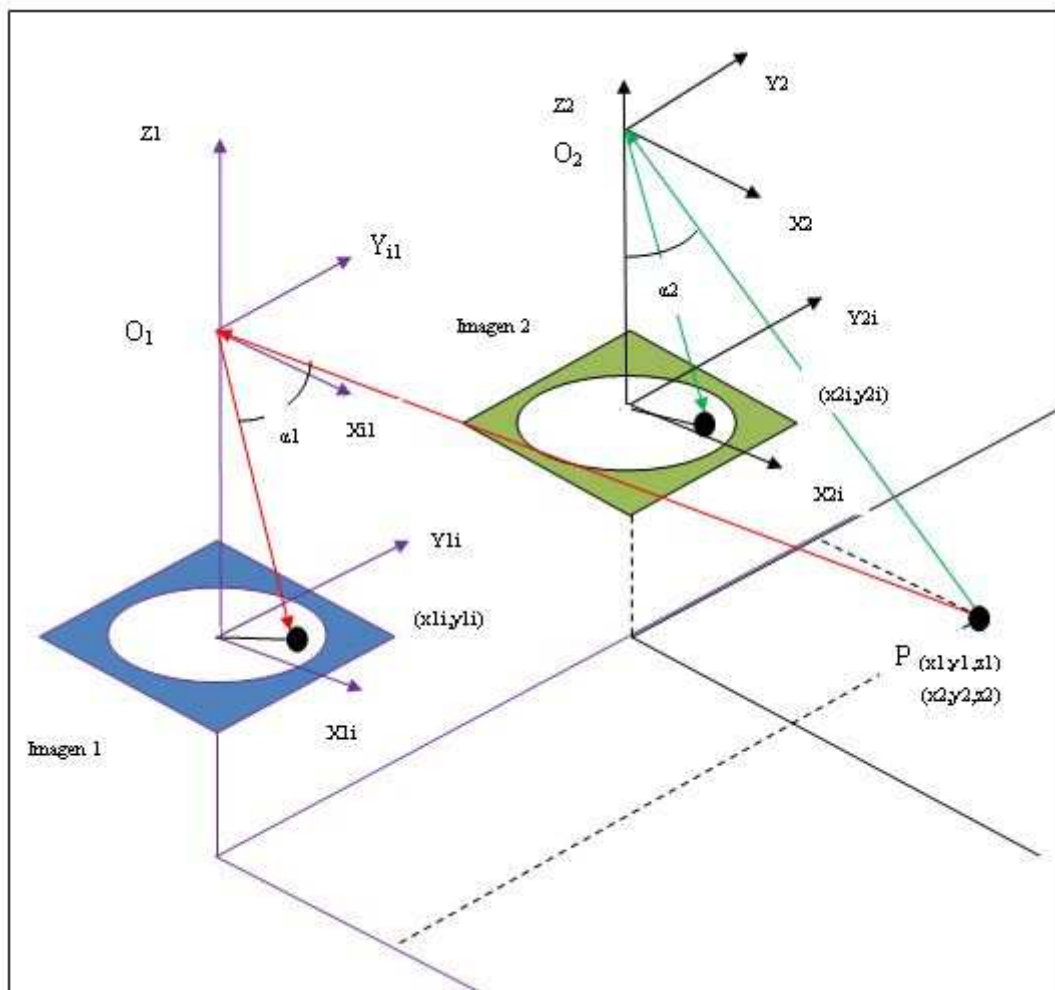


Figura 2.27 Relación geométrica de las coordenadas de un punto objeto y su proyección en el plano imagen de dos cámaras de un sistema estéreo

Dado un punto (x_{1i}, y_{1i}) de la imagen 1, los puntos objeto del espacio cuya proyección coinciden en el mismo, determinan la recta $\overline{O_1 P}$ en el espacio. Por otro lado, aplicando a cada uno de los puntos de la recta $\overline{O_1 P}$ su proyección en la imagen 2, obtendremos una línea que deberemos recorrer para localizar el punto homólogo. Esta línea se llama epipolar y su determinación ayudará a resolver el problema de la correspondencia entre las dos imágenes, restringiendo el problema a la consideración de un único grado de libertad.

La proyección de un punto objeto en la imagen dos viene dada por:

$$\begin{aligned}
x_{2i} &= \frac{\frac{2.R}{\pi} .a \tan \left[\frac{\sqrt{x_2^2 + y_2^2}}{z_2} \right]}{\sqrt{\left(\frac{y_2}{x_2} \right)^2 + 1}} \\
y_{2i} &= \frac{\frac{2.R}{\pi} .a \tan \left[\frac{\sqrt{x_2^2 + y_2^2}}{z_2} \right]}{\sqrt{\left(\frac{x_2}{y_2} \right)^2 + 1}}
\end{aligned} \tag{2.6}$$

En la ecuación (2.6) se ha omitido el factor de distorsión de la lente, por simplicidad. Por otro lado, la transformación entre los sistemas de coordenadas de las cámaras 1 y 2 se puede expresar de forma simple mediante:

$$\begin{aligned}
x_2 &= x_1 \\
y_2 &= y_1 - y_{12} \\
z_2 &= z_1
\end{aligned} \tag{2.7}$$

donde y_{12} representa la separación entre las cámaras 1 y 2. Sustituyendo (2.7) en (2.6) y teniendo en cuenta la ecuación de la recta $\overline{O_1P}$ definida por los posibles puntos objeto cuya proyección es (x_{1i}, y_{1i}) tenemos,

$$\begin{aligned}
\frac{y_1}{x_1} &= \frac{y_{1i}}{x_{1i}} \\
\frac{z_1}{x_1} &= \frac{\sqrt{1 + \left(\frac{y_{1i}}{x_{1i}} \right)^2}}{\tan\left(\frac{\pi}{2R} \sqrt{x_{1i}^2 + y_{1i}^2} \right)}
\end{aligned} \tag{2.8}$$

A partir de lo anterior podremos calcular la línea trazada por los puntos de coordenadas (x_{2i}, y_{2i}) obtenidos al recorrer la recta $\overline{O_1P}$. La línea así obtenida es la epipolar y servirá de guía para determinar la correspondencia de un punto de la imagen 1 en la imagen 2, tal y como se ha mencionado previamente.

2.4.2 Métodos basados en el area

Son métodos que establecen la correspondencia píxel a píxel, utilizando para ello la información, principalmente de intensidad (color) entre un píxel y sus vecinos en una imagen con el píxel y sus vecinos en la otra imagen del par estereoscópico. De este modo, se determina una medida de similitud entre ambos entornos. Uno de los métodos más utilizados para la medida de la similitud consiste en el cómputo del coeficiente de correlación, basado en los mencionados entornos.

Para que la correspondencia pueda establecerse correctamente, se requiere los valores de las intensidades de las imágenes del par estereoscópico sean similares.

En la figura 2.28 se muestran dos imágenes, izquierda y derecha, que constituyen un par estereoscópico. En ellas se han seleccionado una serie de puntos de interés, siendo el objetivo determinar en sendas imágenes cuál es la correspondencia entre ellos. Obsérvese, en este caso la similitud de valores de intensidad entre píxeles homólogos de ambas imágenes.

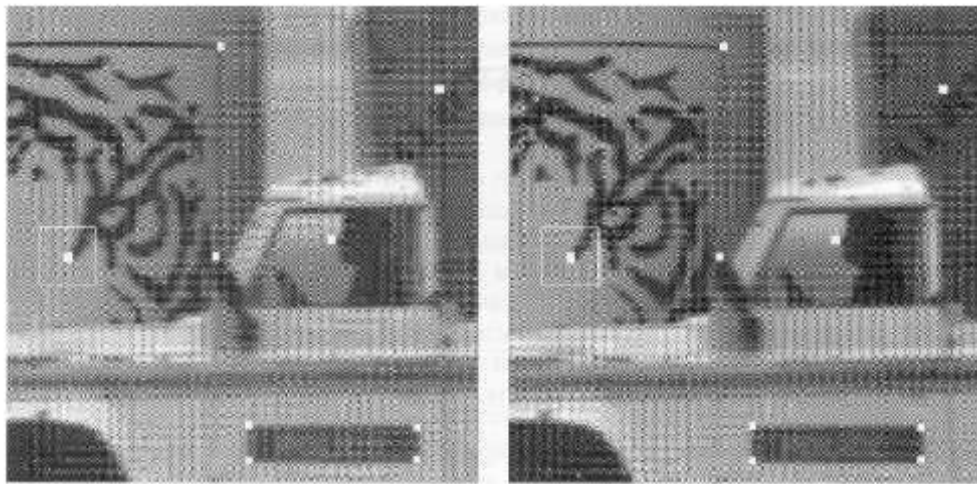


Figura 2.28 Ejemplo de correspondencia de puntos

La estructura básica del algoritmo se describe en la figura 2.29.

Entradas:

- Imagen izquierda (I_i) e imagen derecha (I_d)
- Anchura de la ventana N , siendo $N=2W+1$.
- Región de búsqueda en la imagen derecha $R(p_i)$ asociado con un píxel p_i en la imagen izquierda

Para cada píxel $p_i = (i, j)$ en la imagen izquierda:

- Para cada desplazamiento $\bar{d} = (\bar{d}_1, \bar{d}_2) \in R(p_i)$ calcular la similitud entre la ventana en la imagen izquierda y la ventana en la imagen derecha ($c(d)$)
- La disparidad de p_i es el vector $\bar{d} = (\bar{d}_1, \bar{d}_2)$ que maximiza $c(d)$ sobre $R(p_i)$

$$d = \arg \max_{d \in R} [c(d)]$$

Figura 2.29 Algoritmo de correspondencia basada en el área.

Considerando ventanas de $N \times N$ centradas en los puntos a estudiar y donde $I_i(i, j)$ e $I_d(i, j)$ denotan las intensidades de un píxel en la posición (i, j) pueden considerarse las siguientes funciones como medidas de similitud:

- Correlación cruzada:

$$c(d) = \sum_{k=-W}^W \sum_{l=-W}^W I_i(i+k, j+l) I_d(i+k-d_1, j+l-d_2) \quad (2.9)$$

Normalmente dividiremos (2.9) entre la desviación estándar de I_i y I_d :

$$\bar{c}(d) = \frac{\sum_{k=-W}^W \sum_{l=-W}^W (I_i(i+k, j+l) - \bar{I}_i) (I_d(i+k-d_1, j+l-d_2) - \bar{I}_d)}{\sqrt{\sum_{k=-W}^W \sum_{l=-W}^W (I_i(i+k, j+l) - \bar{I}_i)^2 \sum_{k=-W}^W \sum_{l=-W}^W (I_d(i+k-d_1, j+l-d_2) - \bar{I}_d)^2}} \quad (2.10)$$

- Suma de diferencias absolutas:

$$\bar{c}(d) = - \sum_{k=-W}^W \sum_{l=-W}^W |I_i(i+k, j+l) - I_d(i+k-d_1, j+l-d_2)| \quad (2.11)$$

- Suma de los cuadrados de las diferencias:

$$\bar{c}(d) = - \sum_{k=-W}^W \sum_{l=-W}^W (I_i(i+k, j+l) - I_d(i+k-d_1, j+l-d_2))^2 \quad (2.12)$$

El éxito de los métodos basados en correlación depende de que la ventana utilizada capture la estructura que rodea al punto que se está estudiando para establecer la correspondencia.

¿Cómo elegir el tamaño de la ventana?

Una ventana demasiado pequeña puede no capturar suficiente estructura de la imagen, y puede ser demasiado sensible al ruido. Una ventana demasiado grande es menos sensible al ruido, pero a cambio está sujeta a posibles variaciones de intensidad entre las dos imágenes del par estereoscópico.

Realizar la correlación sobre todos y cada uno de los píxeles de la imagen homóloga requiere un elevado tiempo de computación. Para mejorar el tiempo de respuesta de la correlación añadimos la restricción epipolar, que consiste en limitar la búsqueda sobre el epípolo de la imagen homóloga. El epípolo es el resultado de la proyección en la cámara homóloga de la línea de proyección que causó el píxel a emparejar.

Como ya se comentó, los métodos basados en el área buscan la correspondencia entre las imágenes usando la correlación como medida de similitud. Teniendo en cuenta las restricciones anteriormente indicadas se ha probado la correspondencia a nivel de píxel y a nivel de ventana para aquellas regiones de la imagen pertenecientes a troncos.

Se han probado las medidas de similitud (2.10), (2.11) y (2.12) sobre las regiones de la figura 2.30.

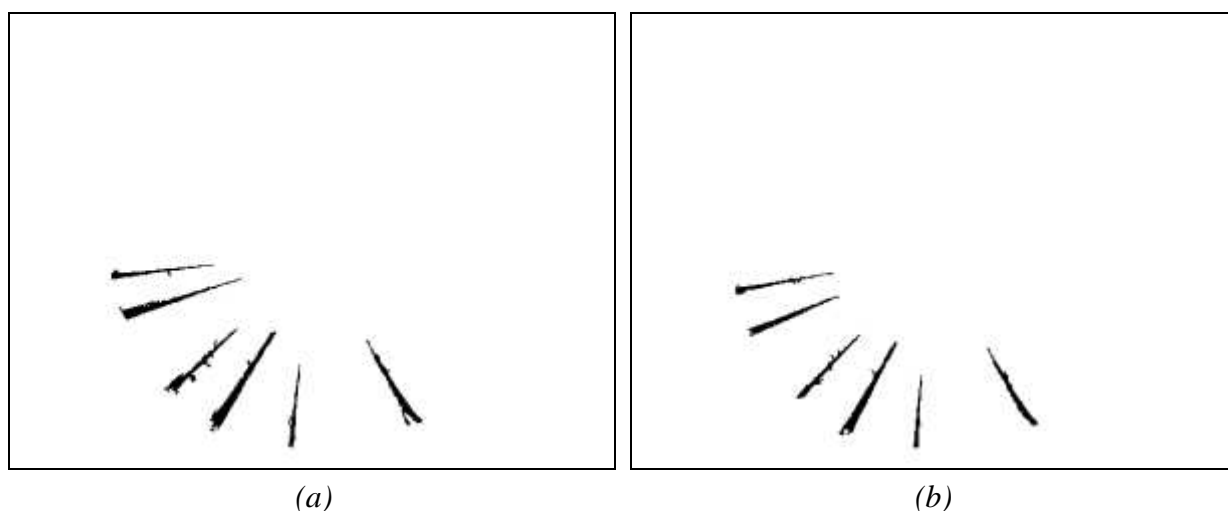


Figura 2.30 Regiones segmentadas a partir de un par de imágenes captadas con un sistema estéreo.

Por cada píxel en la imagen 2.30(a) se busca su homólogo en la imagen 2.30(b) y obtenemos una diferencia de ángulo que se representa con una etiqueta de color. En la figura 2.31 se muestra los rangos de colores utilizados para representar las diferencias de ángulos.



Figura 2.31 Rango de colores utilizados para representar la correspondencia.

Correlación cruzada

En la figura 2.32 pueden verse los resultados obtenidos usando la correlación cruzada (2.10) y usando distintos tamaños de ventana sobre las imágenes de la figura 2.30.

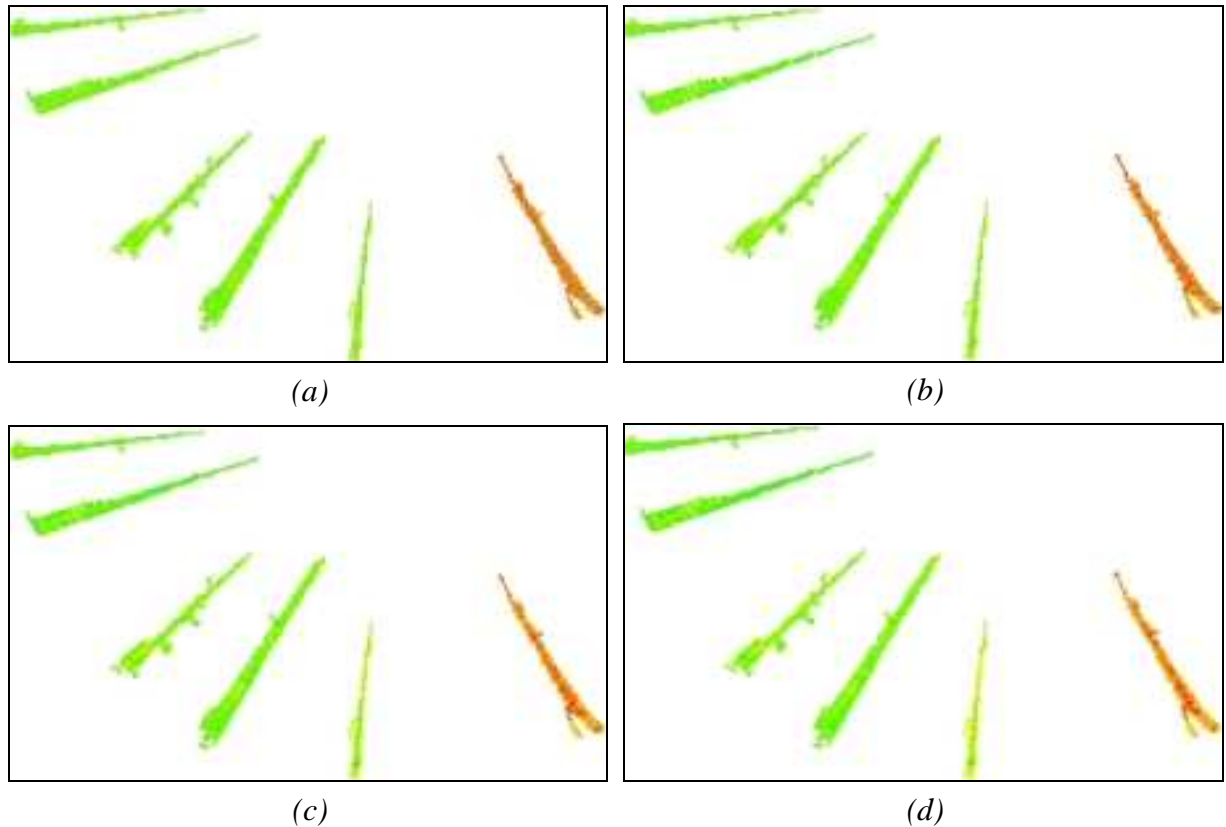


Figura 2.32 Correspondencia basada en el área usando correlación cruzada y distintos tamaños de ventana (N). (a) Correspondencia con $N=3$. (b) Correspondencia con $N=5$. (c) Correspondencia con $N=7$. (d) Correspondencia con $N=11$.

Usando la correlación cruzada como función de similitud no tiene sentido utilizar un tamaño de ventana $N=1$ ya que en ese caso siempre se obtiene como resultado un valor nulo.

Como puede verse en la figura 2.31 las disparidades obtenidas para cada píxel no son continuas. Con esta función de similitud los resultados no son satisfactorios.

Suma de diferencias absolutas

En la figura 2.33 pueden verse los resultados obtenidos usando la suma de diferencias absolutas (2.11) y aplicando distintos tamaños de ventana sobre las imágenes de la figura 2.30.

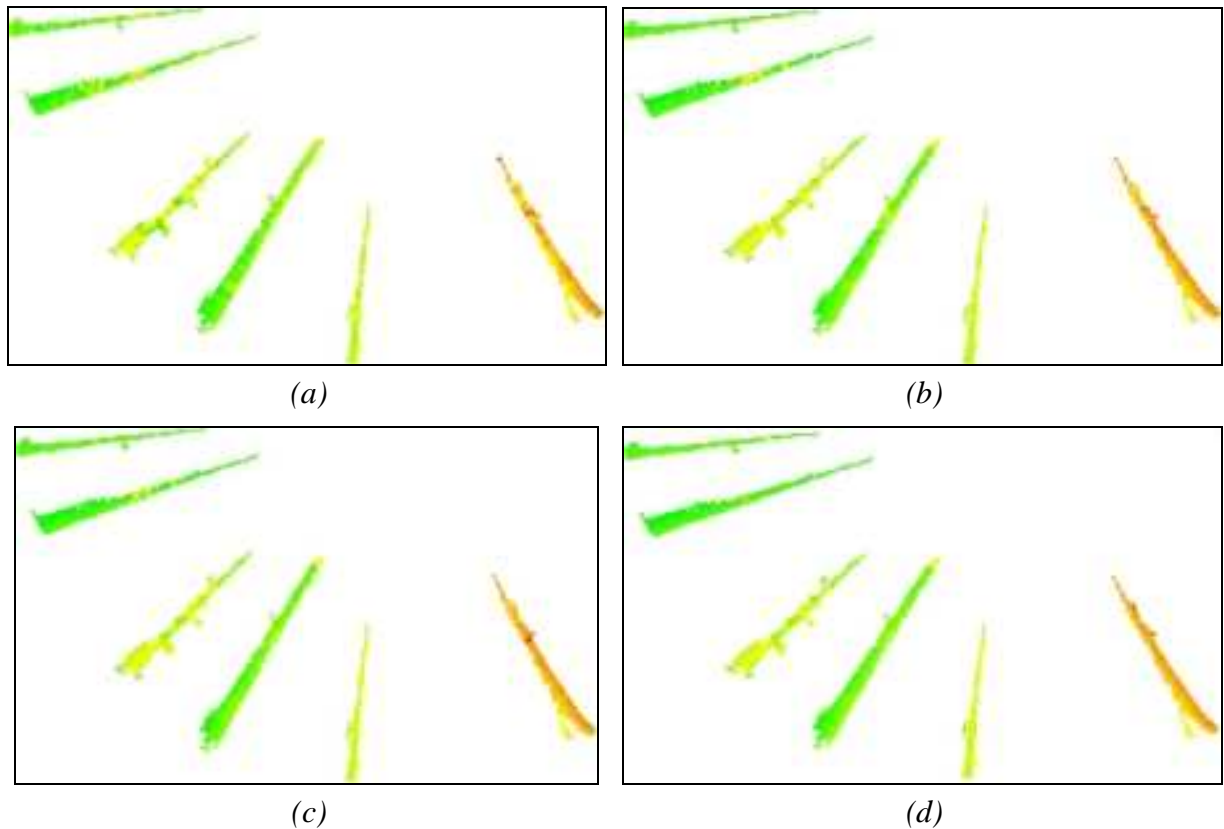


Figura 2.33 Correspondencia basada en el área usando suma de diferencias absolutas y distintos tamaños de ventana (N). (a) Correspondencia con $N=1$. (b) Correspondencia con $N=3$. (c) Correspondencia con $N=5$. (d) Correspondencia con $N=7$.

Cuando se usa un tamaño de ventana $N=1$, se están haciendo comparaciones a nivel de píxel.

Como puede verse en la figura 2.33 las disparidades obtenidas presentan una mayor continuidad que con correlación cruzada salvo en algún tronco particular. Podemos ver que en los árboles situados cerca de 270° , respecto de un sistema de coordenadas centrado en el origen de la imagen, la diferencia de ángulos es cercana a 0 mientras que en ángulos cercanos a 180° la diferencia es cercana a 6° .

Suma de los cuadrados de las diferencias

Por último, en la figura 2.34 pueden verse los resultados obtenidos usando la suma de los cuadrados de las diferencias (2.12) como medida de similitud para distintos tamaños de ancho de ventana.

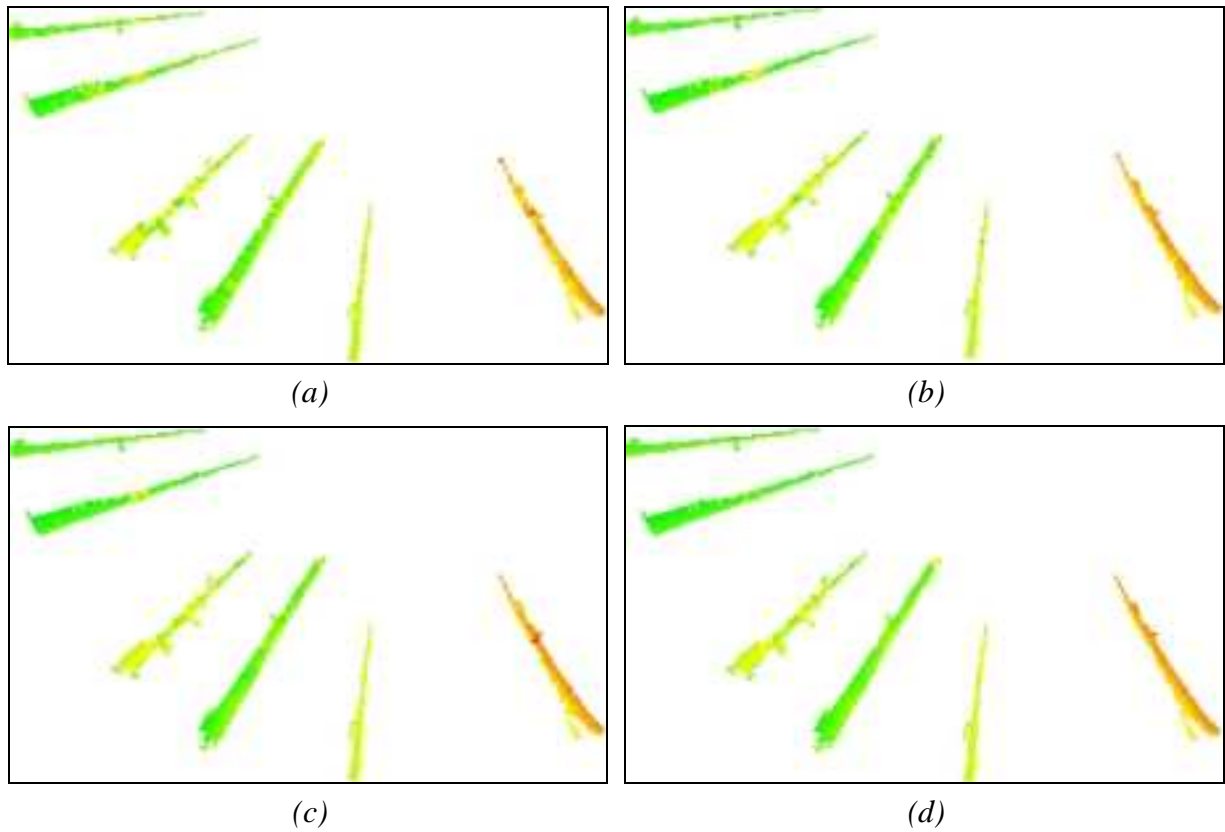


Figura 2.34 Correspondencia basada en el área usando suma de cuadrados de las diferencias y distintos tamaños de ventana (N). (a) Correspondencia con $N=1$. (b) Correspondencia con $N=3$. (c) Correspondencia con $N=5$. (d) Correspondencia con $N=7$.

Los resultados obtenidos con esta función de similitud proporcionan resultados similares a los obtenidos con la suma de diferencias absolutas.

Como puede observarse con tamaños de ventana muy pequeños no se obtienen buenos resultados debido a que no se captura suficiente información de la textura.

Capítulo 3: Diseño de la aplicación

Vistos los fundamentos teóricos de la aplicación, en este capítulo se presenta en primer lugar el diseño a alto nivel de la aplicación, desde el punto de vista de los módulos que la conforman. Desglosándose básicamente en un módulo de procesamiento de imágenes y un segundo módulo relativo a los aspectos relacionados con la presentación al usuario, o lo que es lo mismo la interfaz de usuario. En cada uno de ellos se proporcionan los detalles de diseño más relevantes afectando a cada uno de ellos.

3.1 Módulos

La aplicación se divide en dos grandes módulos:

- Procesamiento de imágenes: se encarga de analizar las dos imágenes que le indica el usuario y de devolver los resultados.
- Presentación al usuario: es la interfaz gráfica gracias a la cual aquél puede elegir las imágenes, además de los datos que quiere que se almacenen tras el análisis, posteriormente puede ver los resultados obtenidos.

El diagrama que explica la relación entre dichos módulos se puede observar en la figura 3.1.

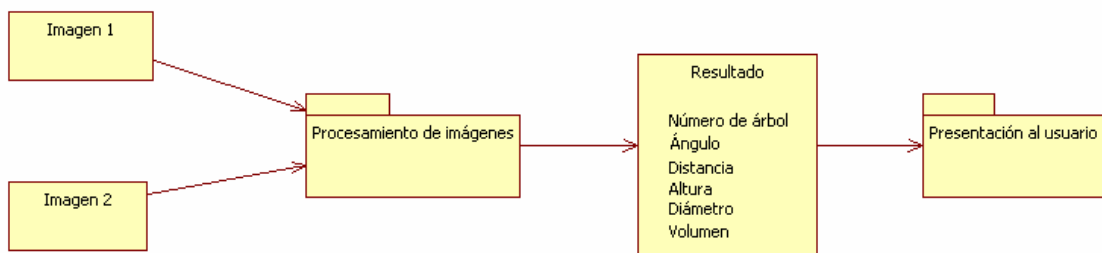


Figura 3.1 Diagrama de la aplicación.

A continuación se proporcionan los detalles pertinentes en relación con el diseño y funcionamiento de cada uno de estos módulos.

3.1.1 Procesamiento de imágenes

Este módulo se encarga, como su nombre indica, del procesamiento de las imágenes a tratar, incluyendo los pasos y procesos necesarios para obtener los resultados básicos del análisis.

Recibe como entrada dos imágenes elegidas por el usuario. Dichas imágenes deben cumplir que posean las mismas dimensiones para que sean admitidas por la aplicación. Además, las imágenes deben ser de la misma zona y capturadas en el mismo instante de tiempo, por lo que deben tener características similares de color. No obstante, en la aplicación que se presenta, este requisito se puede obviar ya que no está conectada al sistema físico de captura y por tanto las imágenes han sido obtenidas previamente, cargándolas en este caso desde los ficheros donde se encuentran almacenadas. Tras la aceptación de las imágenes, se procede a su procesamiento proporcionando a la salida para cada imagen una serie de parámetros asociados a cada uno de los árboles identificados, dichos parámetros son:

- Número de árbol
- Ángulo medido en grados, que determina la posición de cada árbol teniendo en cuenta que el origen del sistema de coordenadas se sitúa en el centro de la imagen
- Diferencia en grados entre un árbol en una de las imágenes del par estereoscópico y el homólogo en la otra imagen
- Altura estimada del tronco en píxeles
- Diámetro estimado del árbol en píxeles

3.1.2 Presentación al usuario

Este módulo contiene la interfaz para poder interactuar con el usuario. El diagrama de diseño de la interfaz se puede observar en la figura 3.2 (omitiendo los atributos y métodos para facilitar su visualización):

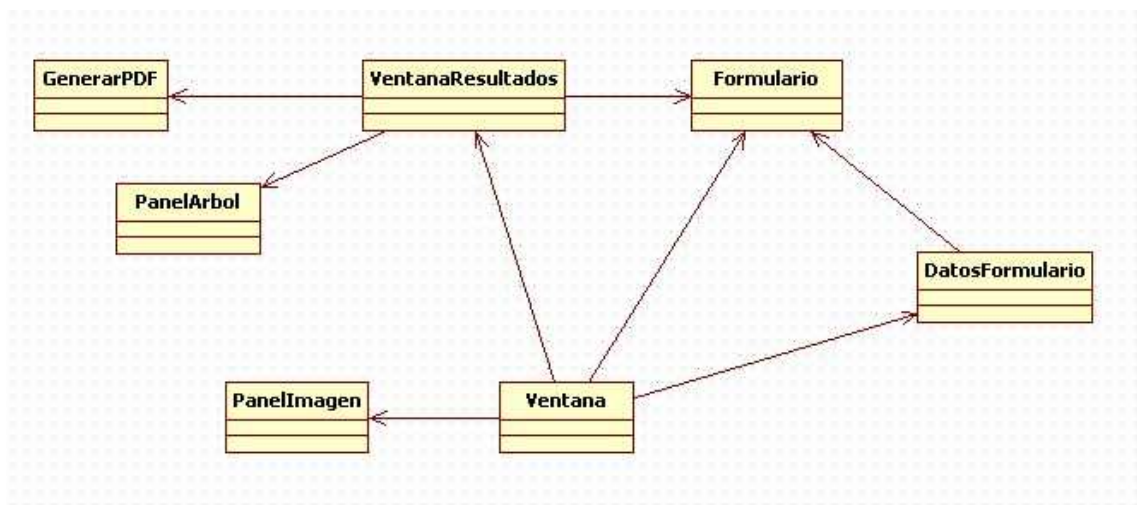


Figura 3.2 Diagrama de diseño de la interfaz.

Recibe como entrada los parámetros asociados a cada uno de los árboles presentes en las imágenes, permitiendo al usuario rellenar un Formulario con los datos que pueda necesitar.

A continuación se explican los aspectos generales de las clases más importantes contenidas en el diagrama de la figura 3.2:

Ventana: es la ventana principal que inicia la aplicación. Desde ella el usuario puede realizar todas las acciones que necesite: elegir las imágenes, rellenar el formulario, elegir los procesos para el posterior análisis y comprobar los resultados.

Formulario: es la ventana desde la cual el usuario rellena los datos del formulario. También se le permite dejar alguno de los datos en blanco, si así lo desea.

VentanaResultados: es la ventana a través de la cual el usuario puede ver los resultados del análisis, así como los datos que ha rellenado en el formulario. Esta ventana únicamente está accesible una vez procesadas las imágenes, ya que antes no tiene ningún sentido. Desde dicha ventana el usuario puede generar un documento en formato PDF para guardar los resultados, también puede visualizar la imagen en tres dimensiones tomando como referencia la imagen original.

En la figura 3.2 se puede observar el diagrama de casos de uso de la aplicación, que son las acciones que puede realizar el usuario. Dichas funciones son:

- **Elegir imágenes:** ambas imágenes tienen que ser de la misma zona, y poseer dimensiones idénticas para poder ser procesadas.
- **Rellenar formulario:** conteniendo los campos vacíos relativos a distintas características sobre la zona donde fueron tomadas las imágenes: latitud, longitud, lugar, fecha y comentarios que el usuario desee añadir.
- **Generar PDF:** si lo desea puede generar un documento en formato PDF con todos los resultados del análisis: imágenes elegidas, algoritmo empleado, datos obtenidos del análisis y datos rellenados en el formulario, para poder guardarlo y usarlo en futuras comparaciones.
- **Ver imagen en tres dimensiones:** a partir de los resultados del análisis, se puede representar el bosque con la librería java3D de manera que se obtiene una imagen, que es fiel reflejo de la imagen original, permitiendo comprobar los resultados del análisis de manera sencilla y eficiente.
- **Seleccionar método:** puede elegir el algoritmo a utilizar de entre los desarrollados en la aplicación para realizar el procesamiento de las imágenes.
- **Observar resultados:** a través de la VentanaResultados, puede observar los resultados del análisis, así como los datos que ha rellenado en el formulario.

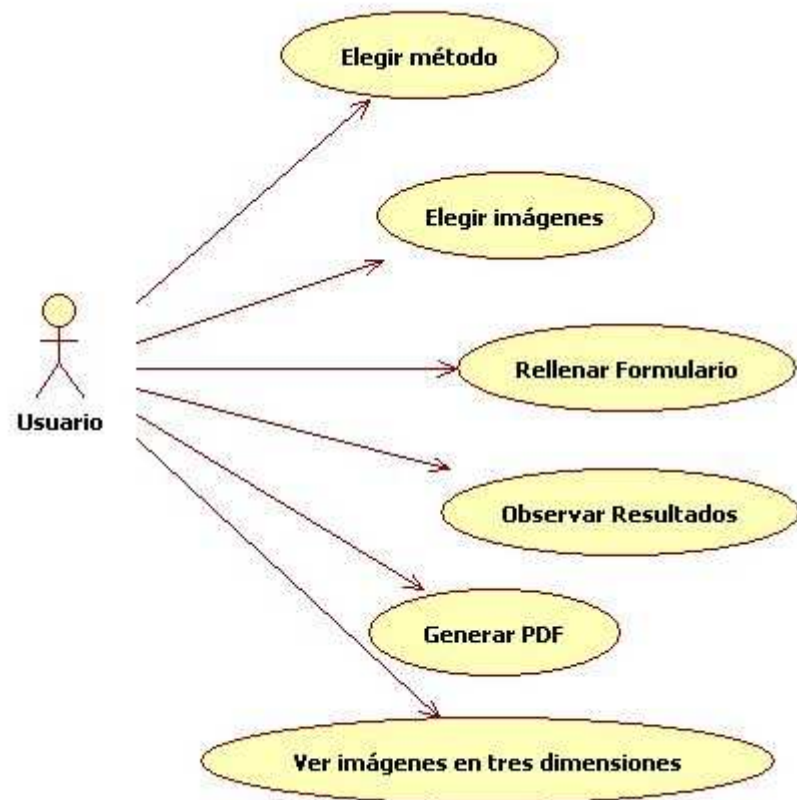


Figura 3.3 Diagrama de casos de uso de la aplicación.

Capítulo 4: Interfaz de usuario

La interfaz de cualquier aplicación es una parte básica de la misma, ya que es la manera que tiene el usuario de comunicarse con la aplicación. Es necesario que sea clara y sencilla de utilizar, con el fin de que el usuario final no tenga ningún problema en su utilización, evitando así los posibles rechazos en el uso de la aplicación.

Además, al tratarse del único procedimiento que posee la aplicación para comunicarse con el usuario, debe garantizar que se puedan realizar todas las acciones que hemos explicado anteriormente. Esto se consigue mediante el diseño e implementación de una serie de menús y de botones con una clara accesibilidad. En este sentido, la característica que prevalece es la fácil identificación de la acción que el usuario trata de realizar.

Teniendo en cuenta el diagrama de diseño de la figura 3.2, así como el diagrama de casos de uso descrito en la figura 3.3, a continuación se explican las partes principales que componen finalmente la interfaz implementada, así como las diferentes ventanas de diálogo que van apareciendo según las acciones seleccionadas.

4.1 Inicio

La aplicación se inicia realmente ejecutando el fichero de tipo “jar”, que la acompaña, requiriéndose tener convenientemente instalada la correspondiente máquina virtual de JAVA.

De este modo, al comenzar la aplicación, el usuario se encuentra con una ventana en la que únicamente se le permite realizar dos acciones, que se describen posteriormente, tal y como se puede observar en la siguiente figura 4.1:

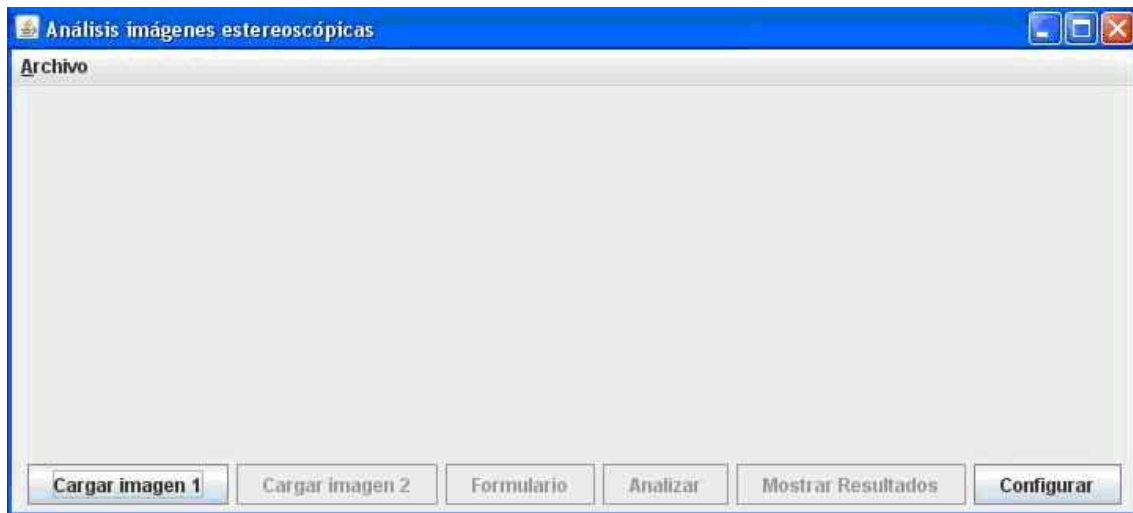


Figura 4.1 Ventana inicial de la aplicación.

Las acciones anteriormente mencionadas y que aparecen en la figura 4.1, permiten al usuario abrir una nueva imagen, pulsando el botón *Cargar imagen 1* o bien elegir el algoritmo que desea utilizar, pulsando *Configurar*. A continuación se explicarán las sucesivas ventanas generadas al seleccionar cada una de las acciones mencionadas.

Podemos observar que la mayoría de los botones se encuentran deshabilitados, esto es así para garantizar que no se puedan realizar ciertas acciones hasta que no se han elegido las dos imágenes, tal y como requiere la especificación de la aplicación. Esto se ha determinado así por el hecho de tratarse de un sistema que procesa un par de imágenes estereoscópicas, en este caso un entorno forestal.

De este modo, por ejemplo, hasta que el usuario no elija las dos imágenes, no le permitimos que realice el análisis, o que rellene el formulario, ya que aún no dispone de los suficientes datos, de esta manera no dejamos que el usuario se precipite y cometa errores a la hora de utilizar la aplicación.

4.2 Configurar

Si el usuario pulsa el botón *Configurar*, se le muestra una ventana como la de la figura 4.2, ofreciéndosele la opción de elegir entre los algoritmos disponibles.

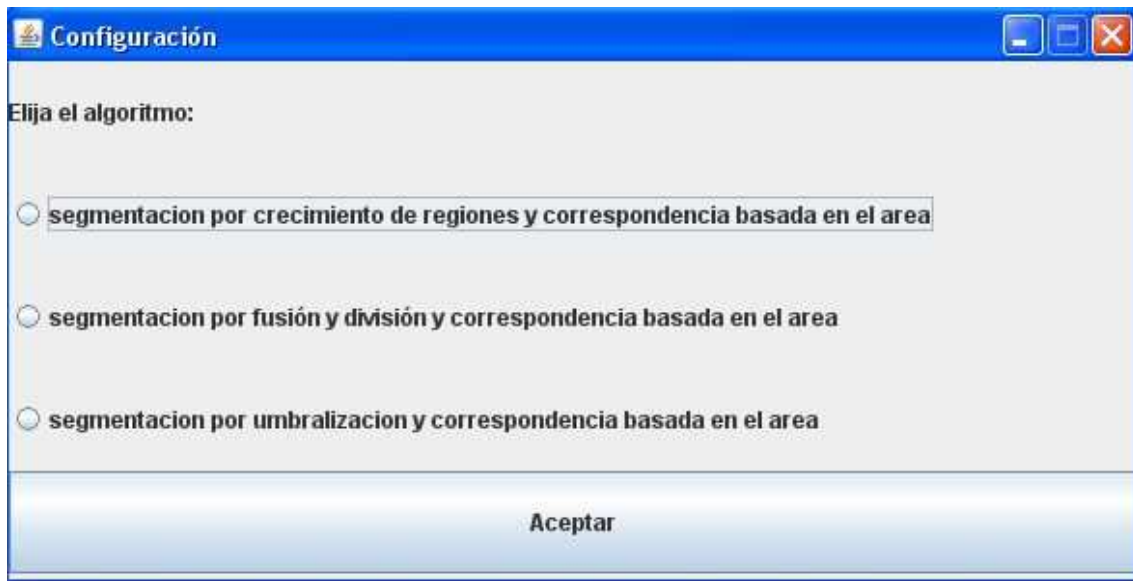


Figura 4.2 Ventana con los algoritmos disponibles en la aplicación

Como se puede observar en la figura anterior, es posible elegir entre los distintos métodos desarrollados en el capítulo 2, que son:

- segmentación por crecimiento de regiones, explicado en el apartado 2.3.2.1 y correspondencia basada en el area (apartado 2.4).
- segmentación por Fusión y División (Split & Merge), explicado en el apartado 2.3.2.2 y correspondencia basada en el area (apartado 2.4).
- segmentación por umbralización, explicado en el apartado 2.3.1 y correspondencia basada en el area (apartado 2.4).

El usuario puede cambiar el algoritmo a utilizar en cualquier momento de la aplicación.

4.3 Cargar imágenes

Al pulsar el botón *Cargar imagen 1* se muestra un diálogo de apertura para que el usuario pueda elegir la primera imagen del par estereoscópico que desea analizar. En la figura 4.3 se puede observar dicho diálogo, de forma que se le proporciona la ruta desde la que se puede cargar la imagen en cuestión.

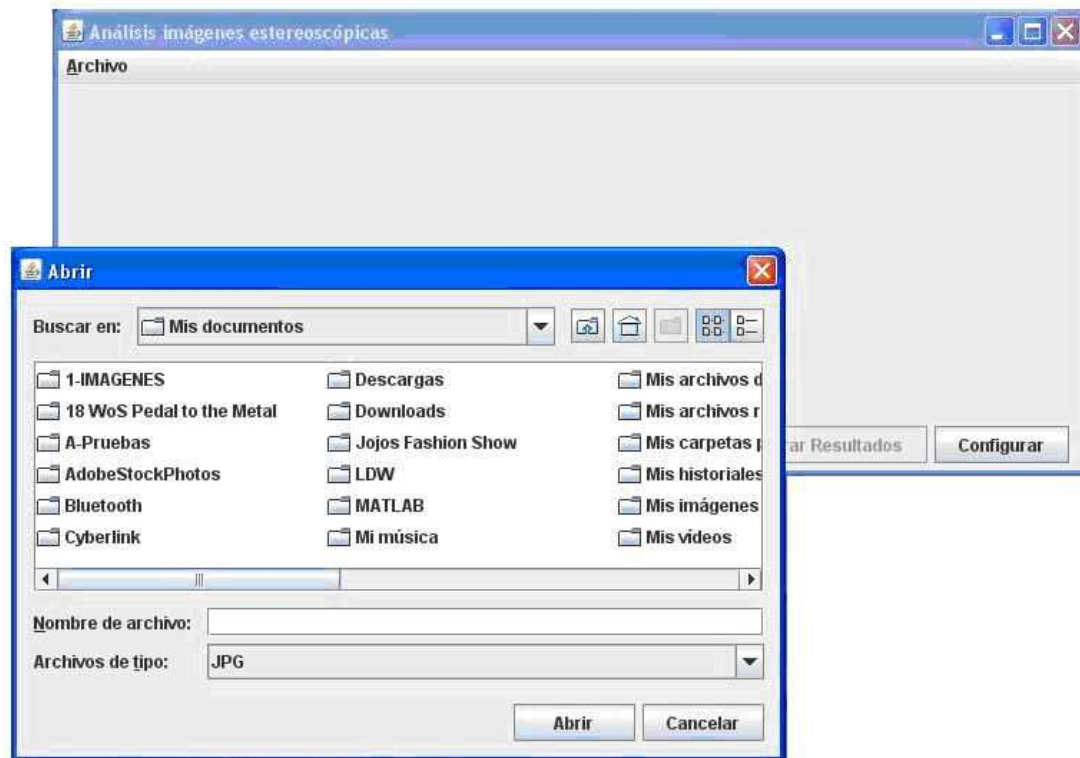


Figura 4.3 Diálogo para elegir la imagen.

Una vez se ha elegido la primera imagen, ésta se muestra en la pantalla, en la parte izquierda, a la espera de que el usuario elija la segunda imagen del par estereoscópico, momento en el cual se habilita la opción *Cargar imagen 2* con el propósito mencionado. La figura 4.4 nos muestra la ventana de diálogo en este momento.



Figura 4.4 Ventana con una imagen elegida.

Una vez que el usuario pulsa el botón *Cargar imagen 2* se le muestra un nuevo diálogo de apertura, tal como aparecía al elegir la primera imagen. Cuando el usuario ha elegido la segunda imagen, ésta se muestra por pantalla y se habilitan nuevos botones como podemos observar en la figura 4.5.

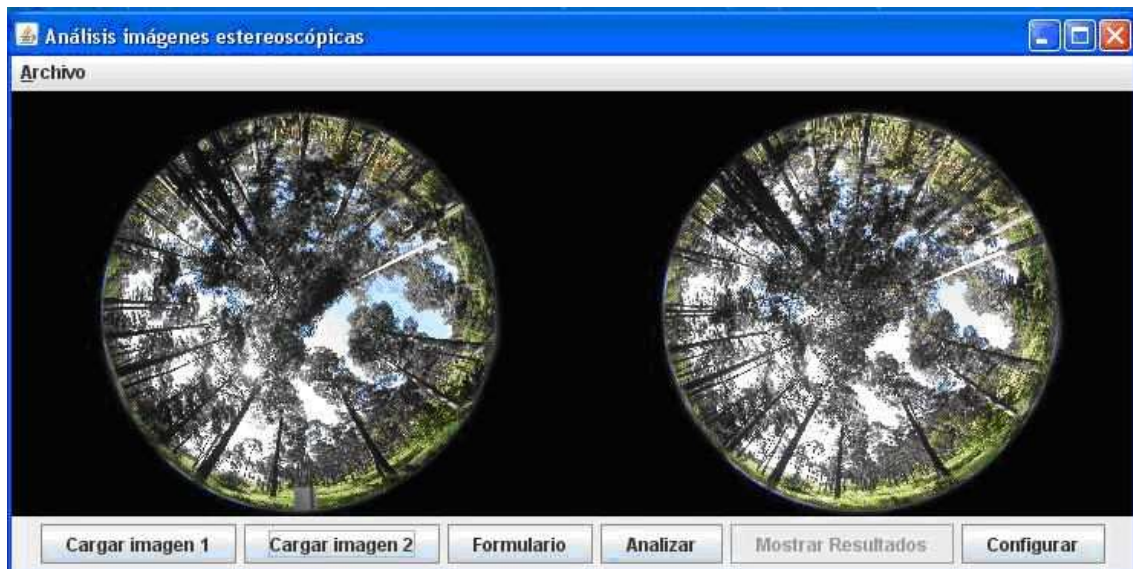


Figura 4.5 Ventana con las dos imágenes elegidas.

En este momento, ya están prácticamente todos los botones disponibles habilitados, el único que todavía permanece inactivo es el botón *Mostrar Resultados*, puesto que los resultados no pueden ser mostrados hasta que no se haya realizado el procesamiento del método o métodos a ejecutar.

Si una vez cargada la primera imagen, el usuario vuelve a pulsar el botón *Cargar Imagen 1*, se procede a preguntar al usuario si desea eliminar todos los datos que tenía hasta ahora. Es una forma de reiniciar la aplicación, esto se ha diseñado de este modo ante el hipotético caso de que el usuario desee probar nuevas imágenes. De este modo, no necesita cerrar la aplicación para posteriormente volverla a abrir. Se trata, pues de una facilidad, que incorporan la mayoría de los programas profesionales dentro del mundo del tratamiento de imágenes, si bien es a su vez una facilidad general, implementada por la mayoría de los programas comerciales software. De esta forma se evita el disponer de una nueva opción tal como *Reiniciar*. La figura 4.6 muestra dicho mensaje:



Figura 4.6 Ventana con el mensaje para reiniciar.

Si el usuario elige la opción *Cancelar*, no se borrará ningún dato, la aplicación continuará desde el punto donde estaba, pero si opta por la opción *Aceptar*, se realizarán las siguientes acciones:

- Se eliminarán las dos imágenes de la ventana.
- Se deshabilitarán nuevamente todos los botones del interfaz.
- Se abrirá un diálogo para que el usuario pueda elegir una nueva imagen.
- Se eliminarán todos los resultados del procesamiento (si éste ya se había realizado).
- Se eliminarán todos los datos del formulario (si el usuario ya lo había rellenado).

Una vez realizadas todas las acciones anteriormente mencionadas, el usuario ya puede elegir la imagen que desea, pudiendo continuar utilizando la aplicación.

4.4 Formulario

Cuando el usuario pulsa el botón *Formulario*, le aparece una nueva ventana a través de la cual se le ofrece la posibilidad de incluir diversos datos que le puedan ser de utilidad en posteriores análisis de resultados. Para ello, previamente habrá necesitado guardar los datos obtenidos en cada procesado. En la figura 4.7 se muestra dicha ventana.

| Formulario | |
|----------------------|----------------------|
| Imagen 1: | p246_1.JPG |
| Imagen 2: | p246_2.JPG |
| Fecha: | <input type="text"/> |
| Coordenadas | |
| Latitud: | <input type="text"/> |
| Longitud: | <input type="text"/> |
| Lugar: | <input type="text"/> |
| Comentarios: | |
| <input type="text"/> | |
| Aceptar | |

Como puede observarse, en el formulario aparecen los nombres de las imágenes elegidas, además de una serie de datos que puede rellenar: fecha, latitud, longitud, lugar y comentarios. Los datos del formulario son voluntarios, esto significa que el usuario puede rellenar todos los datos que desee, de tal forma que si deja algún campo vacío, éste no constará en el informe de resultados y datos generados por la aplicación.

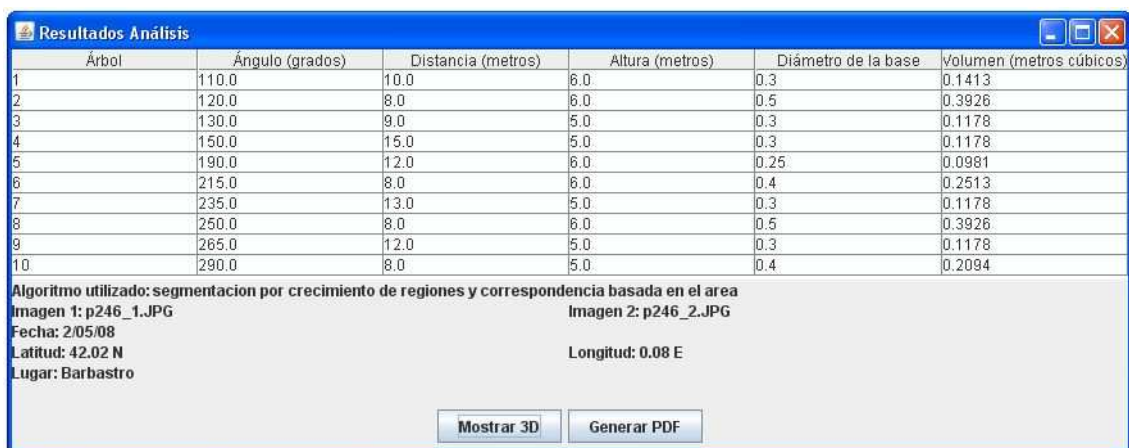
Una vez pulsa la opción aceptar, se guardan automáticamente todos los campos que han sido convenientemente rellenos.

Más adelante, el usuario puede modificar cualquiera de los campos pulsando nuevamente el botón *Formulario*. En ese caso le aparecerán rellenos los campos disponibles que había incorporado con anterioridad, pudiendo añadir nuevos datos, modificar los existentes o incluso borrarlos.

4.5 Análisis

Al pulsar el botón *Analizar* se procede a mostrar los resultados obtenidos a través del proceso o procesos ejecutados. Dependiendo de la complejidad del método o del algoritmo seleccionado, el coste computacional varía, pudiendo llegar a consumir varios minutos. En los casos más costosos se llega a valores del orden de 5 minutos.

Realmente, el coste computacional puede ser un aspecto importante pero no relevante, desde el punto de vista de la aplicación desarrollada. En efecto, el interés se centra en proporcionar al usuario una herramienta que le facilite su tarea. En la actualidad, los usuarios que se encuentran procesando este tipo de imágenes consumen para cada una de ellas del orden de hasta dos horas. Esto significa, que comparado este tiempo con los de ejecución, anteriormente mencionados, aquéllos no son en absoluto relevantes. En cualquier caso, una vez finalizado el proceso, se procede a mostrar los datos sobre la correspondiente ventana, tal y como aparece en la figura 4.8.



| Árbol | Ángulo (grados) | Distancia (metros) | Altura (metros) | Diámetro de la base | Volumen (metros cúbicos) |
|-------|-----------------|--------------------|-----------------|---------------------|--------------------------|
| 1 | 110.0 | 10.0 | 6.0 | 0.3 | 0.1413 |
| 2 | 120.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 3 | 130.0 | 9.0 | 5.0 | 0.3 | 0.1178 |
| 4 | 150.0 | 15.0 | 5.0 | 0.3 | 0.1178 |
| 5 | 190.0 | 12.0 | 6.0 | 0.25 | 0.0981 |
| 6 | 215.0 | 8.0 | 6.0 | 0.4 | 0.2513 |
| 7 | 235.0 | 13.0 | 5.0 | 0.3 | 0.1178 |
| 8 | 250.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 9 | 265.0 | 12.0 | 5.0 | 0.3 | 0.1178 |
| 10 | 290.0 | 8.0 | 5.0 | 0.4 | 0.2094 |

Algoritmo utilizado: segmentacion por crecimiento de regiones y correspondencia basada en el area
Imagen 1: p246_1.JPG Imagen 2: p246_2.JPG
Fecha: 2/05/08
Latitud: 42.02 N Longitud: 0.08 E
Lugar: Barbastro

Mostrar 3D Generar PDF

Figura 4.8 Ventana con los resultados del análisis.

En la figura 4.8 anterior se pueden ver los resultados del análisis. Éstos se muestran en forma de tabla. Cada fila de la tabla representa uno de los troncos de árbol identificados en la imagen encontrado por el algoritmo. A su vez, por cada tronco de árbol identificado con un valor numérico digital en la columna de la izquierda, se muestran

los siguientes datos: el ángulo de posición de cada árbol con respecto al sistema de coordenadas de referencia con origen en el centro de la imagen; la distancia en metros respecto al sistema de cámaras desde la que se tomó la imagen; la altura del árbol en metros; el diámetro de la base del tronco del árbol en metros y finalmente el volumen de madera estimado para dicho árbol a partir de los datos anteriores.

Como puede observarse, en la misma figura 4.8, se muestran también todos los campos del formulario que previamente había rellenado el usuario, así como el algoritmo elegido por el usuario.

Una vez finalizado el procesamiento de las imágenes, el botón *Mostrar Resultados* ya se encuentra disponible, de forma tal que al ser seleccionado, vuelve a visualizarse la misma ventana de la figura 4.8..

Finalmente, en la figura 4.9 podemos observar cómo en este momento del procesamiento, ya se encuentran habilitados todos los botones de la aplicación.

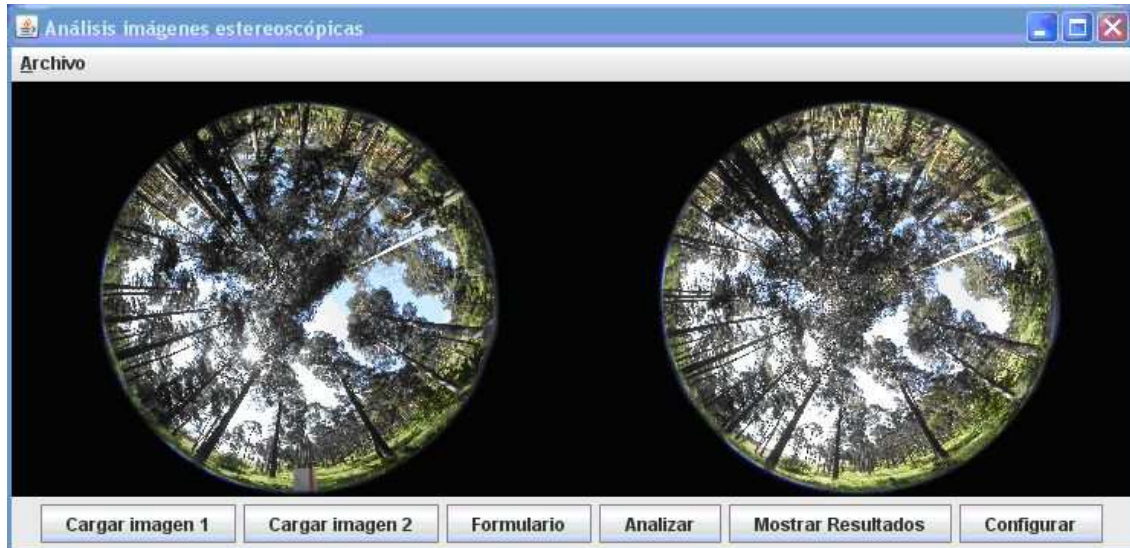


Figura 4.9 Ventana con todos los botones habilitados.

Si posteriormente el usuario modifica alguno de los campos del formulario y vuelve a pulsar el botón *Mostrar Resultados*, los cambios sobre el mismo se verán actualizados en la ventana de resultados correspondiente.

4.5.1 Mostrar 3D

Si el usuario pulsa el botón *Mostrar 3D* aparece una nueva ventana con la imagen de los árboles obtenidos a través de los datos mostrados en la tabla anterior. Estos gráficos se obtienen a partir de las funciones proporcionadas por la librería de java 3D. Obsérvese, no obstante, que en este caso no se identifican todos los árboles existentes en la imagen, esto es debido a que los métodos implementados por la aplicación se encuentran actualmente en estado de investigación en el grupo ISCAR (Ingeniería de Sistemas, Control, Automatización y Robótica), no habiéndose obtenido hasta el momento resultados satisfactorios.



Figura 4.10 A la izquierda se puede observar la imagen original y a la derecha la copia en 3D obtenida con los resultados del análisis.

4.5.2 Generar PDF

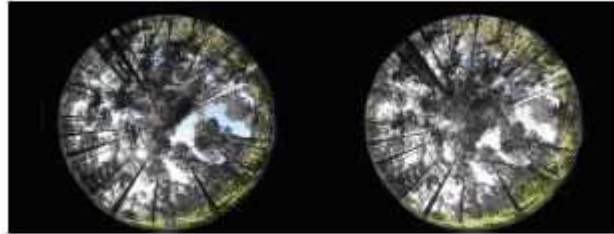
Al pulsar el botón *Generar PDF* aparece un cuadro de diálogo para solicitar al usuario que indique dónde desea almacenar el documento en formato PDF. Una vez ha elegido la ruta, se le muestra el documento tal y como se ha guardado en la ubicación seleccionada.

RESULTADOS DEL ANÁLISIS

Mon Jun 29 14:39:45 CEST 2009

Algoritmo utilizado: segmentacion por crecimiento de regiones y correspondencia basada en el area

Imágenes Iniciales:



Formulario:

- Fecha: 2/05/08
- Latitud: 42.02 N
- Longitud: 0.08 E
- Lugar: Barbastro

Imágenes Resultado:



Tabla Árboles:

| Árbol | Ángulo (grados) | Distancia (metros) | Altura (metros) | Diámetro de la base (metros) | Volumen (metros cúbicos) |
|-------|-----------------|--------------------|-----------------|------------------------------|--------------------------|
| 1 | 110.0 | 10.0 | 6.0 | 0.3 | 0.1413 |
| 2 | 120.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 3 | 130.0 | 9.0 | 5.0 | 0.3 | 0.1178 |
| 4 | 150.0 | 15.0 | 5.0 | 0.3 | 0.1178 |
| 5 | 190.0 | 12.0 | 6.0 | 0.25 | 0.0981 |
| 6 | 215.0 | 8.0 | 6.0 | 0.4 | 0.2513 |
| 7 | 235.0 | 13.0 | 5.0 | 0.3 | 0.1178 |
| 8 | 250.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 9 | 265.0 | 12.0 | 5.0 | 0.3 | 0.1178 |
| 10 | 290.0 | 8.0 | 5.0 | 0.4 | 0.2094 |

Figura 4.11 Documento PDF generado.

Capítulo 5. Conclusiones

En este capítulo se exponen las conclusiones más relevantes obtenidas durante el desarrollo de la aplicación, así como aquellas otras relativas a los resultados obtenidos, junto con las futuras mejoras que se pueden introducir en la aplicación en el futuro, así como las posibilidades de adaptación para incluir nuevas funcionalidades en avances de futuro.

5.1 Resultados

En esta parte se contienen los aspectos relativos a lo que en todo proyecto se identifica como fase de pruebas. Se muestran los resultados obtenidos a través de la aplicación al realizar el procesamiento de dos imágenes estereoscópicas correspondientes a una misma zona boscosa.

Para realizar el análisis de resultados y en línea con lo anterior, el usuario debe elegir dos imágenes que correspondan a la misma región de la escena tridimensional del bosque. Si por ejemplo elige las imágenes de la figura 5.1, desde el punto de vista del observador humano o usuario se puede determinar qué árbol de una imagen, sin pérdida de generalidad llamada imagen izquierda, se corresponde con el árbol de la otra imagen, identificada como imagen derecha. No obstante, desde el punto de vista del computador nos encontramos claramente ante un problema de extrema complejidad. Si bien, desde la perspectiva de las pruebas planteadas, no se trata de analizar la efectividad de los métodos, que como se ha mencionado previamente se encuentran en la actualidad bajo estudio, sino de analizar el comportamiento de la aplicación respecto de la integración de los diferentes módulos y su comportamiento en relación al funcionamiento de cada una de los métodos implementados de forma aislada.

Tampoco se entra en detalles relativos a los tiempos de cómputo, de manera que a la hora de utilizar la aplicación el usuario tendrá en cuenta estas consideraciones en función de la metodología aplicada.

En resumen, se trataría de encontrar un compromiso entre precisión y tiempo computacional, de forma que a juicio del usuario final, éste determine qué prefiere valorar más o menos. No obstante, como se ha indicado previamente, en el desarrollo de la aplicación no se han considerado los tiempos de cómputo de una forma especial, más bien se ha valorado el diseño integrado de diversos módulos y la flexibilidad para futuras ampliaciones. Por este motivo, no se proporcionan tiempos de ejecución.

En la figura 5.1 se presentan dos imágenes estereoscópicas utilizadas en los experimentos. Mientras que en la figura 5.2 se muestran los resultados obtenidos mediante el método de extracción de troncos con la técnica de crecimiento de regiones. Si se comparan las imágenes originales con los resultados obtenidos con la aplicación, se pueden observar algunos de los troncos extraídos. Se puede observar que en este caso la aplicación funciona bastante bien, esto es debido a que los árboles se pueden distinguir muy bien entre sí. Un hecho relevante que destaca es que de todos los troncos posibles sólo se ha conseguido segmentar diez. Ello es debido a que en este caso concreto el método sólo encuentra las regiones de los troncos que se sitúan en posición contraria a la del sol, ya que en la parte donde los troncos son iluminados directamente, las texturas de éstos aparecen bajo diversos aspectos de color.

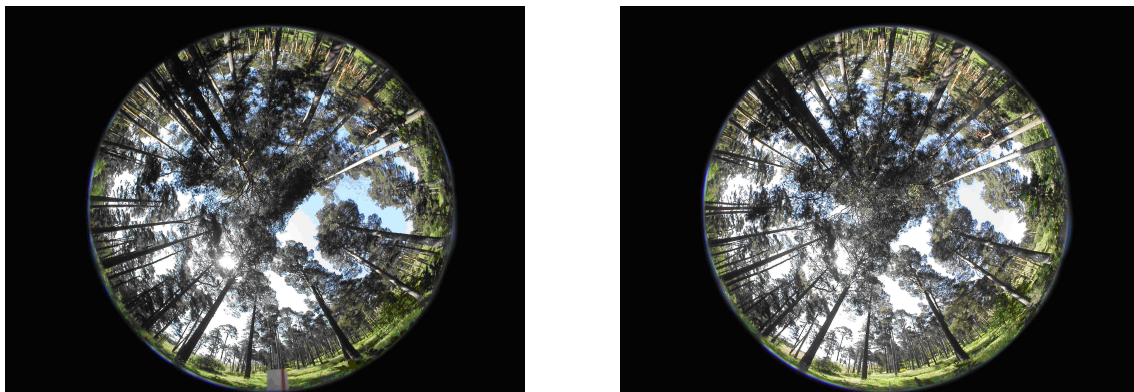


Figura 5.1 Imágenes originales.

Obsérvese, en este caso, cómo algunas partes del tronco aparecen totalmente blancas, otras marrones y otras con tonalidades muy diversas además de abundantes zonas de sombra. Como se ha comentado reiteradamente a lo largo de este trabajo, esto se debe a que los métodos de extracción de regiones bajo este procedimiento se encuentran en fase de investigación.

También se observa que el método sólo localiza los árboles que se sitúan más próximos a la cámara, esto es porque el resto de árboles no interesan. En este caso, únicamente encuentra diez de ellos. Esto es debido a que solamente es posible determinar el volumen con un margen de error aceptable en el caso de este tipo de troncos, ya que para el resto, los errores cometidos son de tal envergadura, que no es posible llegar a un nivel de precisión elevado. Se trata de un requisito impuesto por los propios usuarios finales durante la definición de requisitos de la aplicación.



Figura 5.2 Imágenes resultado obtenidas mediante crecimiento de regiones.

Los parámetros obtenidos para cada uno de los diez árboles identificados son los que se muestran en la tabla de la figura 5.3

| Árbol | Ángulo (grados) | Distancia (metros) | Altura (metros) | Diámetro de la base (metros) | Volumen (metros cúbicos) |
|-------|-----------------|--------------------|-----------------|------------------------------|--------------------------|
| 1 | 110.0 | 10.0 | 6.0 | 0.3 | 0.1413 |
| 2 | 120.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 3 | 130.0 | 9.0 | 5.0 | 0.3 | 0.1178 |
| 4 | 150.0 | 15.0 | 5.0 | 0.3 | 0.1178 |
| 5 | 190.0 | 12.0 | 6.0 | 0.25 | 0.0981 |
| 6 | 215.0 | 8.0 | 6.0 | 0.4 | 0.2513 |
| 7 | 235.0 | 13.0 | 5.0 | 0.3 | 0.1178 |
| 8 | 250.0 | 8.0 | 6.0 | 0.5 | 0.3926 |
| 9 | 265.0 | 12.0 | 5.0 | 0.3 | 0.1178 |
| 10 | 290.0 | 8.0 | 5.0 | 0.4 | 0.2094 |

Figura 5.3 Tabla de árboles encontrados.

Los resultados mostrados en la tabla 5.3 anterior son orientativos. Actualmente en la aplicación no se calculan dichos resultados, sino que están preparados para futuras

ampliaciones. La aplicación desarrollada se ha centrado en distinguir los árboles en ambas imágenes y obtener la correspondencia entre ellas.

5.2. Observaciones

Además de las reflexiones realizadas en las secciones previas, la calidad de los resultados del análisis de las imágenes puede variar según toda una gama de factores, tales como los que se enumeran a continuación:

- calidad de las imágenes
- cantidad de luz
- reflejos producidos por la luz
- diferentes colores de los árboles
- diferentes texturas de los árboles

Por otra parte, a pesar de que para el ojo humano la tarea de identificación de árboles resulta fácil, no lo es así desde el punto de vista de una máquina computacional, que necesita la incorporación de unos procesos añadidos inteligentes de nivel superior, sobrepasando el mero procesamiento.

La observación anterior puede extenderse no sólo en lo que respecta a la identificación de los troncos, sino al establecimiento de la correspondencia estereoscópica, en el sentido de determinar qué tronco de una imagen se corresponde con cuál otro de la otra imagen del par estereoscópico. Así, para el ojo humano dicha correspondencia es una tarea fácil, mientras que para el computador la tarea se torna de una complejidad elevada.

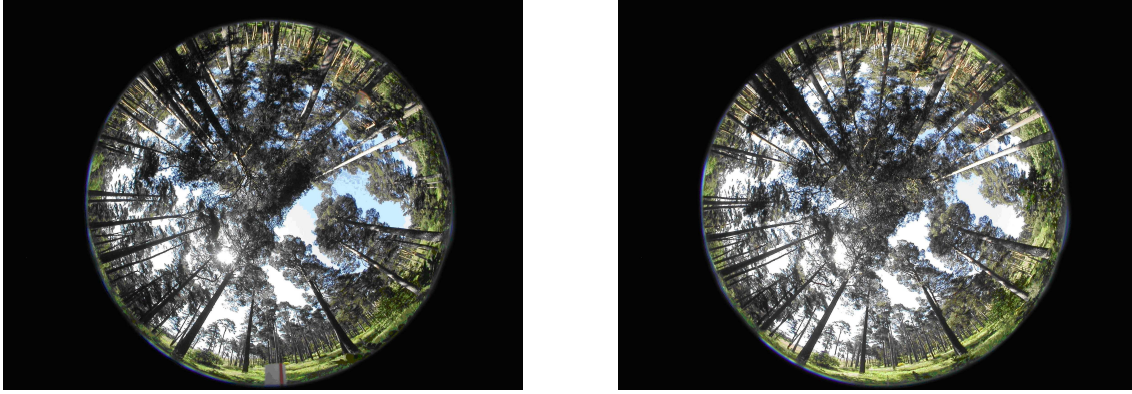


Figura 5.4 Par de imágenes para un posible análisis.

5.3 Futuras líneas de desarrollo

El diseño de la aplicación se ha pensado formando una estructura abierta y flexible, de forma que en el futuro permita la incorporación de nuevas funcionalidades, tanto a nivel de interfaz como de métodos y técnicas de procesamiento.

Debe llevarse a cabo una segunda iteración para que los propios usuarios valoren la aplicación desde las dos perspectivas previstas, esto es, a nivel de interfaz y a nivel de funcionalidad ofrecida respecto de las posibilidades de procesamiento.

Bibliografía

1. Barnea, D.I. and Silverman, H.F. (1972). A Class of Algorithms for Fast Digital image Registration. *IEEE Trans. Computers*, 21, 179-186.
2. Coiras, R.T. Hanson, A.R. Riseman, M.R. and Schlutz, H. (1995). Automatic extraction of buildings and terrain from aerial images. In A. Grün, O. Kübler and P. Agouris (Eds.) *Automatic Extraction of Man Made Objects Aerial and Space Images*, Birkhäuser-Verlag, Basel, pp. 169-178.
3. Davies, E.R. (1997). *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press, London.
4. Darwin, I.F. (2004). *Java Cookbook*, O'Reilly Media, Inc.
5. Fu, K.S., González, R.C. y Lee, C.S.G. (1988). *Robótica: Control, detección, Visión e Inteligencia*. McGraw-Hill, Madrid.
6. Gonzalez, R.C. and Woods, R.E. (1993). *Digital Image Processing*, Addison-Wesley, Reading, MA.
7. Maravall, D. (1993). *Reconocimiento de Formas y Visión Artificial*. RA-MA.
8. Otsu, N. (1979). A threshold selection method from gray-level-histograms. *IEEE Trans. System Man and Cybernetics*. SMC-9, 62-66.
9. Pajares, G. and de la Cruz, J.M. (2007). *Visión por Computador: Imágenes Digitales y Aplicaciones*, RA-MA.
10. Romero, M. A. (2008). *Calibración de cámaras estereoscópicas con lentes de "ojo de pez" para análisis de texturas forestales*. Programa de doctorado Ingeniería de Sistemas y Automática, Proyecto del periodo de investigación.

11. Schwalbe, E., (2005). Geometric Modelling and Calibration of Fisheye Lens Camera Systems. Proc. 2nd Panoramic Photogrammetry Workshop, Int. Archives of Photogrammetry and Remote Sensing, Vol. 36, Part 5/W8,
12. Shirai, Y. (1987). Three-dimensional Computer Vision. Springer-Verlag, Berlín.
13. Sonka, M., Hlavac, V. and Boyle, R. (1995). *Image Processing, Analysis and Machine Vision*, Chapman-Hall, Cambridge.
14. Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall, Upper Saddle River: NJ.

